USBインターフェース
32IN/32OUT　DIOボード

HUSB－DIO464
HUSB－DIO464(D)

ユーザーズマニュアル

ＮＣボードシリーズ
絶縁型入出力ボード

株式会社ハイバーテック
http://www.hivertec.co.jp/

この説明書は

ＤＩＯ４６４・・ＵＳＢ　ｂｕｓ

ＨＵＳＢ－ＤＩＯ４６４

ＨＵＳＢ－ＤＩＯ４６４（Ｄ）に適応しています.

本書及びプログラムの全部又は一部の無断転載，コピーを禁止します.
本製品の内容に関しましては，改良等により将来予告なしに変更することがあります.
本製品の内容についてお気づきの点がございましたら，お手数ながら当社までご連絡下さい.

Windows95，Windows98，WindowsNT 4.0，Windows2000，WindowsXP Home Edition，WindowsXP Professional，VisualC++，Visual Basic，Microsoft C/C++ は Microsoft Corporation の米国及びその他の国における登録商標です.
その他，記載されている会社名，製品名は，各社の商標又は登録商標です.

株式会社 ハイバーテック
東京都墨田区両国４－８－１
ダイユービル
ＴＥＬ　０３－３８４６－３８０１
ＦＡＸ　０３－３８４６－３７７３
sales@hivertec.co.jp

第１．０版 ２００３年　１月２８日発行
不許複製・転載

保証範囲

１．本製品の保証期間は，お買い上げ頂いた日より３年間です．保証期間中に弊社の判断により欠陥が判明した場合には，本製品を弊社に引き取り，修理または交換を行います．

２．保証期間内外に関わらず，弊社製品の使用，供給（納期）または故障に起因する，お客様及び第三者が被った，直接，間接，２次的な損害あるいは，遺失利益の損害に付いて，弊社は本製品の販売価格以上の責任を負わないものとしますので，予めご了承下さい．

免責事項

１．本マニュアルに記載された内容に沿わない，製品の取り付け，接続，設定，運用により生じた損害に対しましては，一切の責任を負いかねますので，予めご了承下さい．

２．本製品は，一般電子機器用（工作機械・計測機器・ＦＡ／ＯＡ機器・通信機器等）に製造された半導体製品を使用していますので，その誤作動や故障が直接，生命を脅かしたり，身体・財産等に危害を及ぼす恐れのある装置（医療機器・交通機器・燃焼機器・安全装置等）に適用できるような設計，意図，または，承認，保証もされていません．
ゆえに本製品の安全性，品質および性能に関しては，本マニュアル（またはカタログ）に記載してあること以外は明示的にも黙示的にも一切保証するものではありませんので，予めご了承下さい．

３．保証期間内外に関わらず，お客様が行った弊社の承認しない製品の改造または，修理が原因で生じた損害に対しましては，一切の責任を負いかねますので，予めご了承下さい．

４．本マニュアルに記載された内容について，弊社もしくは，第三者の特許権，著作権，商標権，その他の知的所有権の権利に対する保証または実施権の許諾を行うものではありません．
また本マニュアルに記載された情報を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合，弊社は，その責任を負いかねますので，予めご了承下さい．

安全にお使い頂くために

この度は，弊社ＮＣボードシリーズをご採用頂きまして，誠に有り難う御座います．本書は，本製品をご使用して頂く場合の取り扱い，留意点に付いて記入してありますので，必ずご一読の上ご利用をお願い致します．
尚，本マニュアルは，本書が添付されたＮＣボード常設箇所付近の分かりやすい場所に常時保管し，必要に応じて適宜参照・確認頂きますよう，お願い致します．

安全上の注意

本製品のご使用前に，必ずこのユーザーズマニュアル及び付属書類を全て熟読し，内容を理解してから正しくご使用下さい．
本製品の知識，安全の情報及び注意事項の全てに付いて習熟してからご使用下さい．
本ユーザーズマニュアルでは，安全注意事項のランクを「警告」，「注意」として 区分してあります．

⚠ **警 告**	この表示を無視して，誤った取り扱いをすると，人が死亡または重傷を負う可能性が想定される内容を示しています．
⚠ **注 意**	この表示を無視して，誤った取り扱いをすると，人が傷害を負う可能性または物的損害が想定される内容を示しています．

１．対象ユーザー

注 意

本製品およびマニュアルは，以下の様な，ユーザーを対象としています.

・拡張用ボードの増設および配線に付いて基本的な知識を有している方.
・制御用電子機器およびパソコン等について基本的な知識を有している方.

２．適合ＢＵＳ

注 意

本製品は Universal Serial Bus 1.1 に適合したボードです.

３．環境条件

警 告

本製品は，下記の環境条件下で保管・ご使用下さい.

・動作周囲温度　０℃～＋５０℃
・動作周囲湿度　２０%ＲＨ～８５%ＲＨ（結露せぬこと）
・保存周囲温度　－１５℃～＋７５℃
・保存周囲湿度　１０%ＲＨ～９０%ＲＨ（結露せぬこと）
・雰囲気　腐食性ガス・引火性ガス・オイルミスト・塵埃のないこと
・標高　海抜３０００m以下
（３００m毎に2℃の上限値を下げた範囲で使用して下さい）

４．運搬・取り付け

警 告

本製品にふれる前に，金属に触り身体の静電気を取り除いて下さい.
静電気は，本ボードの故障の原因になります.

本製品を静電気の帯びやすい梱包材(エアーキャップなど)でくるまないで下さい.
静電気は，本ボードの故障の原因になります.

本製品の上に重いものを載せないで下さい．重いものを乗せますと，部品が損傷し故障の原因になります.

本製品のジャンパー及びディップスイッチの設定は，パソコン等に接続する前に行って下さい.
電源がＯＮの状態で設定しますと，設定を正しく認識しないで誤動作の原因になります.

本製品のジャンパー及びディップスイッチの設定は，正しく行って下さい．設定を間違えますと誤動作の原因になります.

ＵＳＢコネクタは本来 ホットプラグインが許されていますが制御用の接続には電源ＯＮ状態でのＵＳＢコネクタの抜き挿しは避けてください.

本製品をパソコン等と接続する時は，コネクタを深くしっかりと挿入し，ケーブルの直近部分を固定する等して，動作中に抜けたり，接触不良が発生しない様な措置を施して下さい.
動作中に抜けたり，接触不良が発生したりすると，誤動作，故障の原因となります.

注 意

本製品を落とすなど乱暴に扱わないで下さい．衝撃や振動が故障の原因となります.

本製品の半田面を手で直接触らないで下さい.
部品の突起などにより怪我をする恐れがあります.

５．配　線

警 告

外線用コネクタへの配線作業や外線用コネクタの着脱は，パソコン等の電源をＯＦＦにし，電源コードを抜いてから行って下さい.
電源コードを抜かないで作業を行った場合，故障の原因になります.
また，装置が思わぬ動作をすることがあります.

外線用コネクタへの配線は，コネクタ信号表などをよく確認し，正しく配線して下さい.
間違った配線をしますと，故障・焼損の原因になります.

外部から供給する電源は，必ず定格以内でご使用下さい．定格以外で使用されますと，故障・焼損・誤動作の原因となります.

入出力回路に接続する回路は，必ず定格電流・電圧以内でご使用下さい.
定格以外で使用されますと，故障・焼損・誤動作の原因となります.

外部配線用コネクタは，推奨のコネクタをご使用下さい.
推奨以外のコネクタを使用されますと，接触不良などにより誤動作の原因となります.

外部配線用コネクタは，必ずロックしてご使用下さい.
ロックしないで使用されますと，コネクタが外れる，または接触不良などにより誤動作の原因となります.

外部配線用ケーブルは，引っ張る，または重い荷重を掛けないで下さい.
コネクタが外れる，または接触不良などにより誤動作の原因となります.

外部配線用ケーブルは，モーターの配線やＡＣ電源ケーブルなど，ノイズの多い配線とは出来るだけ離して下さい．配線が近いとノイズが 誤動作の原因となります.

６．試運転・調整

警 告

本製品を使用し装置を動作させる時は，プログラムのデバッグを充分行ってから動作させて下さい．プログラムに間違いがありますと，思わぬ動きをすることがあります.

本製品に添付してあるサンプルプログラムを使用し装置を動作させる時，最初は十分に安全な条件で，また機械系に合った設定を行って動作を確認して下さい.
機械系に合わない設定で動作を行うと思わぬ動きをすることがあります.

７．廃　棄

警 告

本製品を廃棄する時は，関連する法律・規則に従って処理して下さい.

目　　次

１．はじめに　・・・

図　表　目　次

１．　はじめに　・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・　１

２．ハードウェア編　・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・　２
表２．　１－１　ＨＵＳＢ－ＤＩＯ４６４仕様　・・・・・・・・・・・・・・・・・・・・　２
写真２．２－１　ＨＵＳＢ―ＤＩＯ４６４　・・・・・・・・・・・・・・・・・・・・・　３
図２．　２－１　ボード形寸　・・・・・・・・・・・・・・・・・・・・・・・・・・・　４
図２．　２－２　ＤＩＮレール取付台形寸　・・・・・・・・・・・・・・・・・・・・・　４
図２．　３－１　ＨＵＳＢ－ＤＩＯ４６４ブロックダイア　・・・・・・・・・・・・・・　５
図２．　４－１　出力回路　・・・・・・・・・・・・・・・・・・・・・・・・・・・・　５
図２．　４－２　入力回路　・・・・・・・・・・・・・・・・・・・・・・・・・・・・　５
表２．　５－１　ポート構成　・・・・・・・・・・・・・・・・・・・・・・・・・・・　６
図２．　６－１　ジャンパ設定個所　・・・・・・・・・・・・・・・・・・・・・・・・　７
図２．　６－２　Ｐ１ジャンパ　・・・・・・・・・・・・・・・・・・・・・・・・・・　７
図２．　６－３　ボードＩＤの設定　・・・・・・・・・・・・・・・・・・・・・・・・　７
表２．　７－１　出力コネクタ（Ｊ１）ピン配列　・・・・・・・・・・・・・・・・・・　８
表２．　７－２　入力コネクタ（Ｊ２）ピン配列　・・・・・・・・・・・・・・・・・・　９
表２．　７－３　ＴＢ１電源端子　・・・・・・・・・・・・・・・・・・・・・・・・・　９
図２．　８－１　出力回路接続例　・・・・・・・・・・・・・・・・・・・・・・・・・　１０
図２．　８－２　シンクタイプとの接続　・・・・・・・・・・・・・・・・・・・・・・　１０
図２．　８－３　外部接続における対策　・・・・・・・・・・・・・・・・・・・・・・　１１

３．ソフトウェア編　・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・　１２
図３．　３－１　デバイスドライバインストール後の確認画面　・・・・・・・・・・・・　１４
図３．　３－２　ＷｉｎＸＰのインストール手順　・・・・・・・・・・・・・・・・・・　１５
図３．　３－３　Ｗｉｎ２Ｋのインストール手順　・・・・・・・・・・・・・・・・・・　１６
図３．　３－４　Ｗｉｎ９８のインストール手順　・・・・・・・・・・・・・・・・・・　１７
図３．　４―１　ＷｉｎＸＰドライバの更新　・・・・・・・・・・・・・・・・・・・・　２０
図３．　４―２　Ｗｉｎ２Ｋドライバの更新　・・・・・・・・・・・・・・・・・・・・　２１
図３．　４―３　Ｗｉｎ９８ドライバの更新　・・・・・・・・・・・・・・・・・・・・　２２
表３．　７―１　関数一覧　・・・・・・・・・・・・・・・・・・・・・・・・・・・・　２６
表３．　７―２　関数通信時間　・・・・・・・・・・・・・・・・・・・・・・・・・・　２６

１．　はじめに

このマニュアルはＵＳＢ 適合の ディジタル入出力ボードＨＵＳＢ－ＤＩＯ４６４及びＨＵＳＢ－ＤＩＯ４６４（Ｄ）の取扱説明書です．（※）

※ＨＵＳＢ－ＤＩＯ４６４（Ｄ）ボードはＨＵＳＢ－ＤＩＯ４６４ボードにＤＩＮレール取付台が追加されたボードです．以降，この説明書では，双方ともにＨＵＳＢ－ＤＩＯ４６４またはＤＩＯ４６４と呼びます．

■ ＨＵＳＢ－ＤＩＯ４６４は　ＵＳＢ １．１　規格に基づいています．

１．１　このマニュアルについて

このマニュアルには次の内容が記載されています．

■　ハードウェアに関する情報
（１）ポート構成
（２）ボード上の設定
（３）コネクタ割付
（４）外部との接続

■　添付ソフトウェア スタートアップガイド
（１）添付ソフトのインストール方法
（２）ドライバＩ／Ｆ用ＤＬＬ関数説明
（３）サンプルプログラム説明

１．２　添付ソフトウェア

このボードには次のソフトウェアが添付されます．

Ｗｉｎｄｏｗｓ版
デバイスドライバ（Ｉ／Ｆ用ＤＬＬ）
サンプルプログラム・・・・・・・・・・ドライバ関数の使用法を解説するサンプルソフトです．

２．ハードウェア編

２．１　ＨＵＳＢ－ＤＩＯ４６４及びＨＵＳＢ－ＤＩＯ４６４（Ｄ）仕 様

項　　目		仕　　様
【 入力部 】	入力点数	３２点
	入力形式	フォトカプラによる絶縁入力（ＴＬＰ２８０－４相当品を使用）
	定格入力電圧	ＤＣ２４Ｖ
	使用入力電圧範囲	ＤＣ２１．６Ｖ～ＤＣ２６．４Ｖ
	定格入力電流	５ｍＡ
	入力抵抗	４．４ｋΩ
	応答時間	１ｍｓｅｃ以内（注１）
	入力点数/コモン数	３２点／１コモン
	入力論理	入力のフォトカプラＯＮで内部論理“１”
【 出力部 】	出力点数	３２点
	出力形式	フォトカプラによる絶縁，オープンコレクタ・トランジスタアレー出力（ＴＤ６２０８４ＡＰ相当品を使用）
	定格負荷電圧	ＤＣ２４Ｖ
	使用負荷電圧範囲	ＤＣ２１．６Ｖ～ＤＣ２６．４Ｖ
	最大負荷電流	８０ｍＡ
	応答時間	１ｍｓｅｃ以内（注２）
	出力点数/コモン数	３２点／１コモン
	出力論理	内部論理“１”で出力のトランジスタがＯＮ
【 周囲条件 】	供給電源	＋２４Ｖ±１０％
	消費電流（約）	本体０．３５ｍＡ ＋ 入力ＯＮ点数×５ｍＡ ＋ 出力ＯＮ点数×８０ｍＡ
	温度条件	０℃～５０℃　ただし，結露ないこと．
	形　　寸	ボードのみ　　：１６０ｍｍ Ｘ １０７．５ｍｍ Ｘ　高さ 約２９ｍｍ
		ＤＩＮ取付台込み：１６１．５ｍｍ Ｘ １１１ｍｍ Ｘ　高さ 約５３ｍｍ

注１．入力コネクタピンに信号が入力されてから，信号がボード内部で認識されるまでの時間
（但し入力ポートのフィルタ設定がＯＦＦの時）

注２．ボード内部で出力が実行されてから，信号が出力コネクタピンに出力されるまでの時間

表２．１－１　ＨＵＳＢ－ＤＩＯ４６４仕様

２．２　ＨＵＳＢ－ＤＩＯ４６４及びＨＵＳＢ－ＤＩＯ４６４（Ｄ）のボード構造

ＵＳＢコネクタの抜き挿しの注意
ＵＳＢコネクタは本来 ホットプラグインが許されていますが制御用の接続には電源ＯＮ状態でのＵＳＢコネクタの抜き挿しは避けてください.

２．２．１　ＨＵＳＢ－ＤＩＯ４６４（Ｄ）呼称

■ ＨＵＳＢ－ＤＩＯ４６４（Ｄ）　の各部の呼称を　写真２．２－１ に示します.

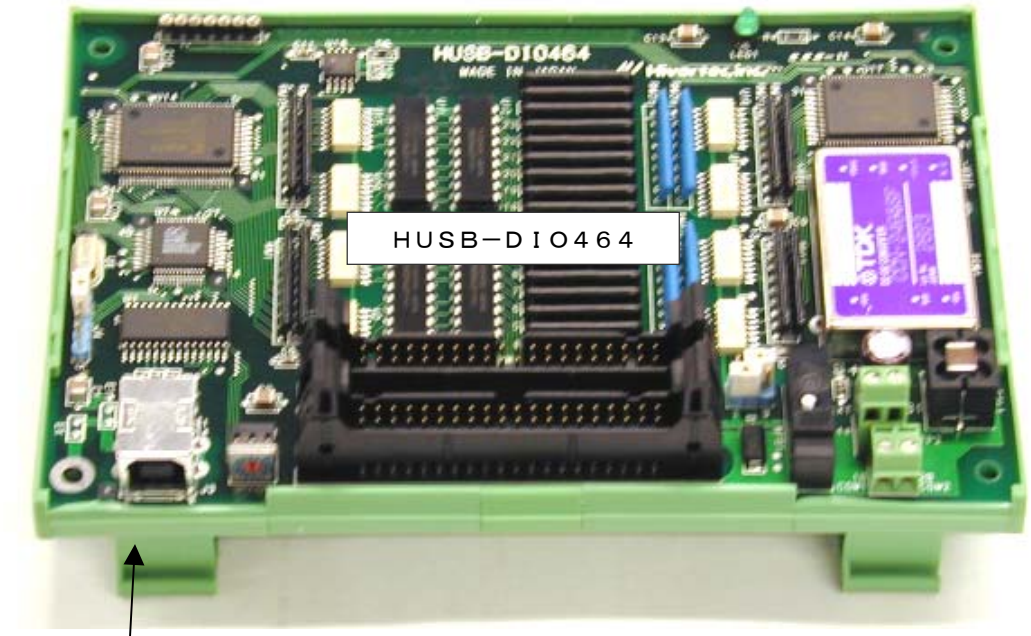


写真２．２－１　ＨＵＳＢ－ＤＩＯ４６４（Ｄ）

２．２．２　ボード形寸

（１）ＨＵＳＢ－ＤＩＯ４６４ボード形寸

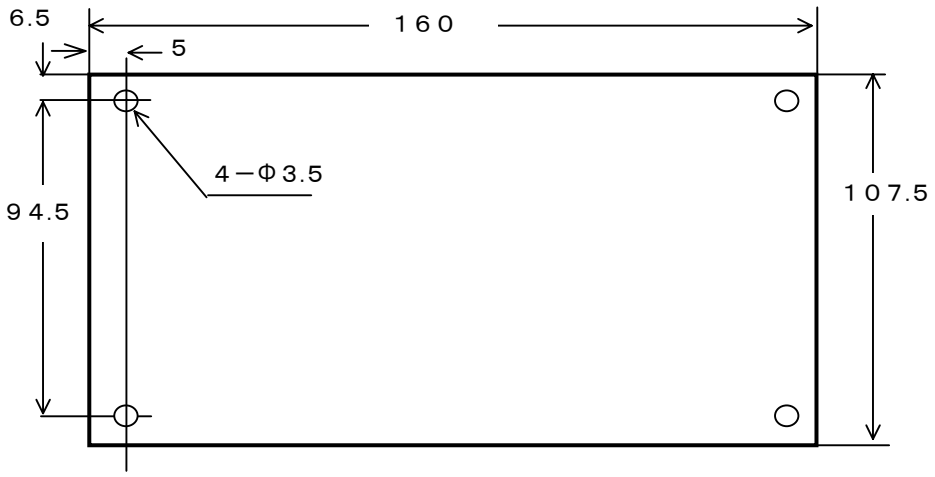


図２．２－１　ボード形寸

（２）ＤＩＮ取付台形寸

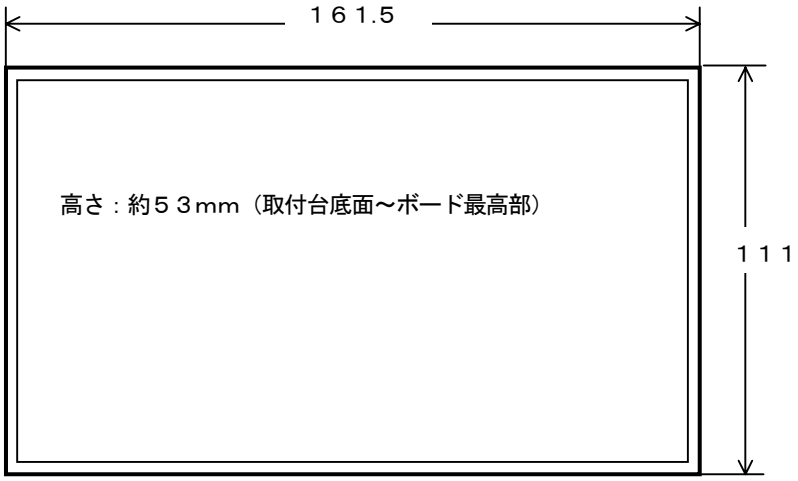


図２．２－２　ＤＩＮレール取付台形寸

２．３　ブロック図

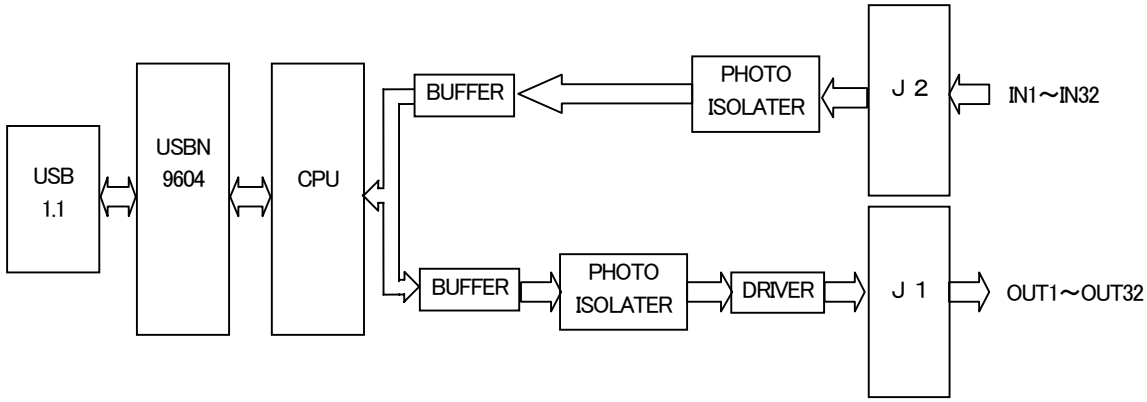


図２．３－１　ＨＵＳＢ―ＤＩＯ４６４ブロックダイア

２．４　入出力回路

（１）本ボードの出力回路を図２．４-１に示します.

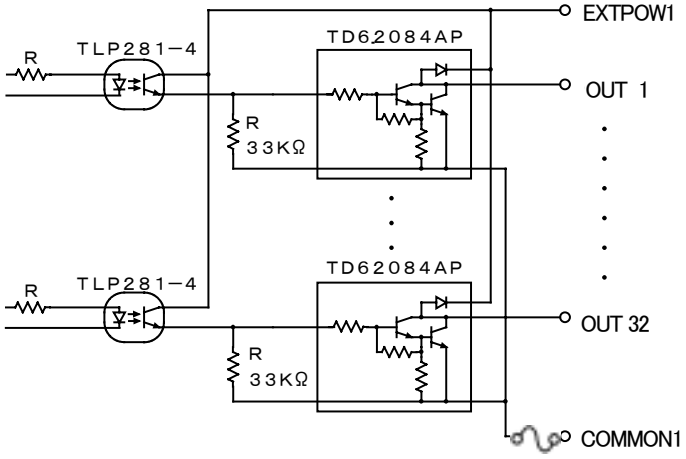


図２．４－１　出力回路

（２）本ボードの入力回路を図２．４-２に示します.

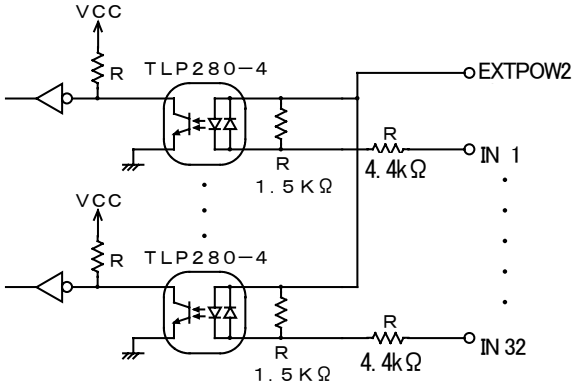


図２．４－２　入力回路

２．５　ポート構成

本ボードのポート構成を表２．５－１に示します.

ポート名	ビット	入力（ＩＮ）
入力ポート１	0	IN 1
	1	IN 2
	2	IN 3
	3	IN 4
	4	IN 5
	5	IN 6
	6	IN 7
	7	IN 8
入力ポート２	0	IN 9
	1	IN10
	2	IN11
	3	IN12
	4	IN13
	5	IN14
	6	IN15
	7	IN16
入力ポート３	0	IN17
	1	IN18
	2	IN19
	3	IN20
	4	IN21
	5	IN22
	6	IN23
	7	IN24
入力ポート４	0	IN25
	1	IN26
	2	IN27
	3	IN28
	4	IN29
	5	IN30
	6	IN31
	7	IN32

ポート名	ビット	入力（ＩＮ）	出力（ＯＵＴ）
出力ポート１	0	OUT 1	OUT 1
	1	OUT 2	OUT 2
	2	OUT 3	OUT 3
	3	OUT 4	OUT 4
	4	OUT 5	OUT 5
	5	OUT 6	OUT 6
	6	OUT 7	OUT 7
	7	OUT 8	OUT 8
出力ポート２	0	OUT 9	OUT 9
	1	OUT10	OUT10
	2	OUT11	OUT11
	3	OUT12	OUT12
	4	OUT13	OUT13
	5	OUT14	OUT14
	6	OUT15	OUT15
	7	OUT16	OUT16
出力ポート３	0	OUT17	OUT17
	1	OUT18	OUT18
	2	OUT19	OUT19
	3	OUT20	OUT20
	4	OUT21	OUT21
	5	OUT22	OUT22
	6	OUT23	OUT23
	7	OUT24	OUT24
出力ポート４	0	OUT25	OUT25
	1	OUT26	OUT26
	2	OUT27	OUT27
	3	OUT28	OUT28
	4	OUT29	OUT29
	5	OUT30	OUT30
	6	OUT31	OUT31
	7	OUT32	OUT32

表２．５－１　ポート構成

２．６　ボード上の設定

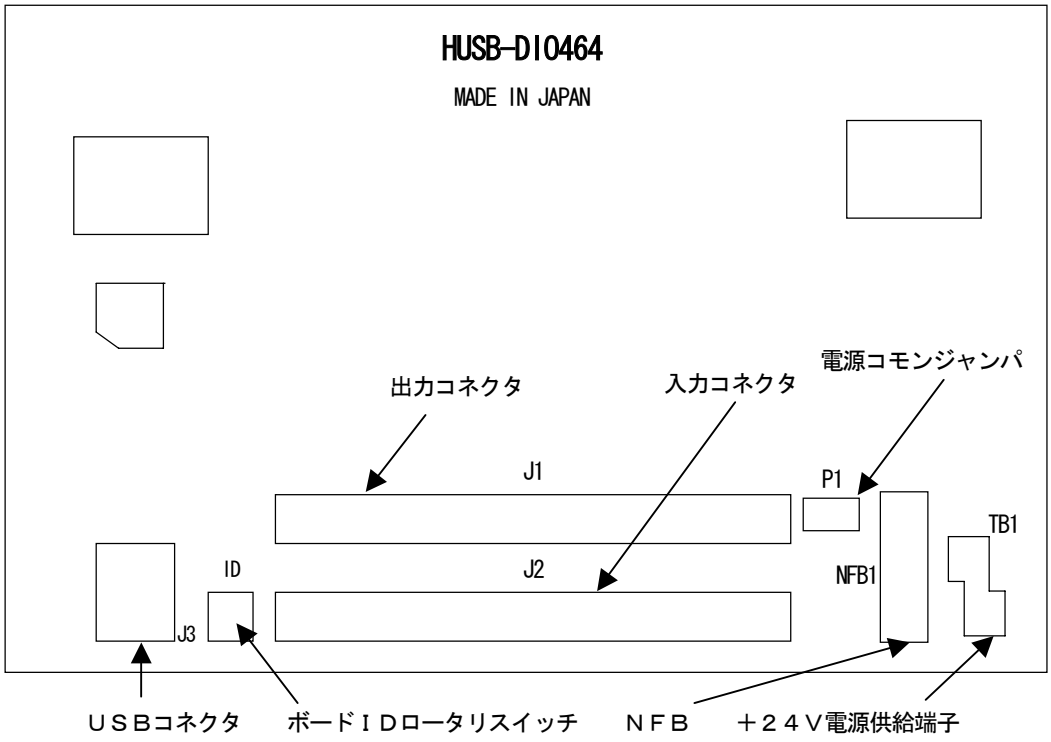


図２．６－１　ジャンパ設定個所

（１）ＴＢ１電源端子およびＰ１ジャンパ

ＴＢ１ は＋２４Ｖ電源供給端子です.

■供給電源：＋２４Ｖ　±１０%

Ｐ１ジャンパは ＋２４Ｖカプラ絶縁電源 ＥＸＴＰＯＷ１，ＥＸＴＰＯＷ２を接続します.

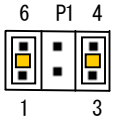
Ｐ１ジャンパの出荷時の設定は１－６接続，３－４接続です.

（COMMON1 と COMMON2 共通，EXTPOW1 と EXTPOW2 共通）

ＴＢ１接続は　「表２．７－３　ＴＢ１電源端子」参照

■ＮＦＢは ＋２４Ｖ逆接続時過電流保護用.（１Ａ）

過電流検出により白色突起が突出します．原因除去後突起押し込みにより復帰します.



COMMON1 と COMMON2 共通，EXTPOW1 と EXTPOW2 共通(出荷時の設定)

図２．６－２　Ｐ１ジャンパ

（２）ボードＩＤロータリスイッチの設定

ボードＩＤは　ＨＵＳＢ－ＤＩＯ４６４ ボードを識別するＮｏ．です.

矢印の位置がＩＤ設定値（１６進数：０～Ｆ(15)）となります．出荷時の設定は“０”になっています.

図２．６－３　ボードＩＤの設定（本図は“Ｆ”設定例）

２．７　コネクタ信号表

（1）出力コネクタ（Ｊ１）

出力コネクタピン配列表を表２．７－１に示します.

ピン番号	信　号　名		ピン番号	信　号　名	
１	EXTPOW1	+24V 入力	２	EXTPOW1	+24V 入力
３	OUT 1	出力 1	４	OUT 2	出力 2
５	OUT 3	出力 3	６	OUT 4	出力 4
７	OUT 5	出力 5	８	OUT 6	出力 6
９	OUT 7	出力 7	１０	OUT 8	出力 8
１１	OUT 9	出力 9	１２	OUT10	出力 10
１３	OUT11	出力 11	１４	OUT12	出力 12
１５	OUT13	出力 13	１６	OUT14	出力 14
１７	OUT15	出力 15	１８	OUT16	出力 16
１９	COMMON1	24V 用コモン（GND）	２０	COMMON1	24V 用コモン（GND）
２１	EXTPOW1	+24V 入力	２２	EXTPOW1	+24V 入力
２３	OUT17	出力 17	２４	OUT18	出力 18
２５	OUT19	出力 19	２６	OUT20	出力 20
２７	OUT21	出力 21	２８	OUT22	出力 22
２９	OUT23	出力 23	３０	OUT24	出力 24
３１	OUT25	出力 25	３２	OUT26	出力 26
３３	OUT27	出力 27	３４	OUT28	出力 28
３５	OUT29	出力 29	３６	OUT30	出力 30
３７	OUT31	出力 31	３８	OUT32	出力 32
３９	COMMON1	24V 用コモン（GND）	４０	COMMON1	24V 用コモン（GND）

注意. EXTPOW1 は，出力ポート用外部絶縁電源供給端子です.

Ｊ１の COMMON1（１９，２０，３９，４０ピン）はすべて接続して下さい.

コネクタ型式：４０ＰＩＮフラットケーブルコネクタ　ＸＧ４Ａ－４０３１　（オムロン）

ケーブル側コネクタ（参考）　：ＸＧ４Ｍ－４０３０－Ｔ　　（フラットケーブル用）

：ＸＧ５Ｍ－４０３２－Ｎ　　（バラ線圧接用）

表２．７－１　出力コネクタ（Ｊ１）ピン配列

（2）入力コネクタ（Ｊ２）

入力コネクタピン配列表を表２．７－２に示します.

ピン番号	信　号　名		ピン番号	信　号　名	
１	EXTPOW2	+24V 入力	２	EXTPOW2	+24V 入力
３	IN 1	入力 1	４	IN 2	入力 2
５	IN 3	入力 3	６	IN 4	入力 4
７	IN 5	入力 5	８	IN 6	入力 6
９	IN 7	入力 7	１０	IN 8	入力 8
１１	IN 9	入力 9	１２	IN10	入力 10
１３	IN11	入力 11	１４	IN12	入力 12
１５	IN13	入力 13	１６	IN14	入力 14
１７	IN15	入力 15	１８	IN16	入力 16
１９	COMMON2	24V 用コモン（GND）	２０	COMMON2	24V 用コモン（GND）
２１	EXTPOW2	+24V 入力	２２	EXTPOW2	+24V 入力
２３	IN17	入力 17	２４	IN18	入力 18
２５	IN19	入力 19	２６	IN20	入力 20
２７	IN21	入力 21	２８	IN22	入力 22
２９	IN23	入力 23	３０	IN24	入力 24
３１	IN25	入力 25	３２	IN26	入力 26
３３	IN27	入力 27	３４	IN28	入力 28
３５	IN29	入力 29	３６	IN30	入力 30
３７	IN31	入力 31	３８	IN32	入力 32
３９	COMMON2	24V 用コモン（GND）	４０	COMMON2	24V 用コモン（GND）

注意. EXTPOW2 は, 入力ポート用外部絶縁電源供給端子です.

コネクタ型式：４０ＰＩＮフラットケーブルコネクタ　ＸＧ４Ａ－４０３１　（オムロン）
ケーブル側コネクタ（参考）　：ＸＧ４Ｍ－４０３０－Ｔ　（フラットケーブル用）
：ＸＧ５Ｍ－４０３２－Ｎ　（バラ線圧接用）

表２．７－２　入力コネクタ（Ｊ２）ピン配列

（3）＋２４Ｖ電源供給端子配列（ＴＢ１）

＋２４Ｖ電源供給端子配列を 表２．７－3に示します.

端子番号	信　号　名		備　　考
1A	EXTPOW1	（＋２４V 外部電源供給）	COMMON1 と COMMON2 はｼﾞｬﾝﾊﾟＰ１の３－４接続により共通になります.
1B	COMMON1	（同ＧＮＤ）	
2A	EXTPOW2	（＋２４V 外部電源供給）	同様に EXTPOW1 と EXTPOW2 はＰ１の１－６接続により共通になります.
2B	COMMON2	（同ＧＮＤ）	

表２．７－３　ＴＢ１ 電源端子

２．８　外部との接続

以下に入出力回路の一般的な接続例を図示します。

２．８．１　出力回路接続例

本ボードの出力回路の接続例を図２．８－１　に示します.

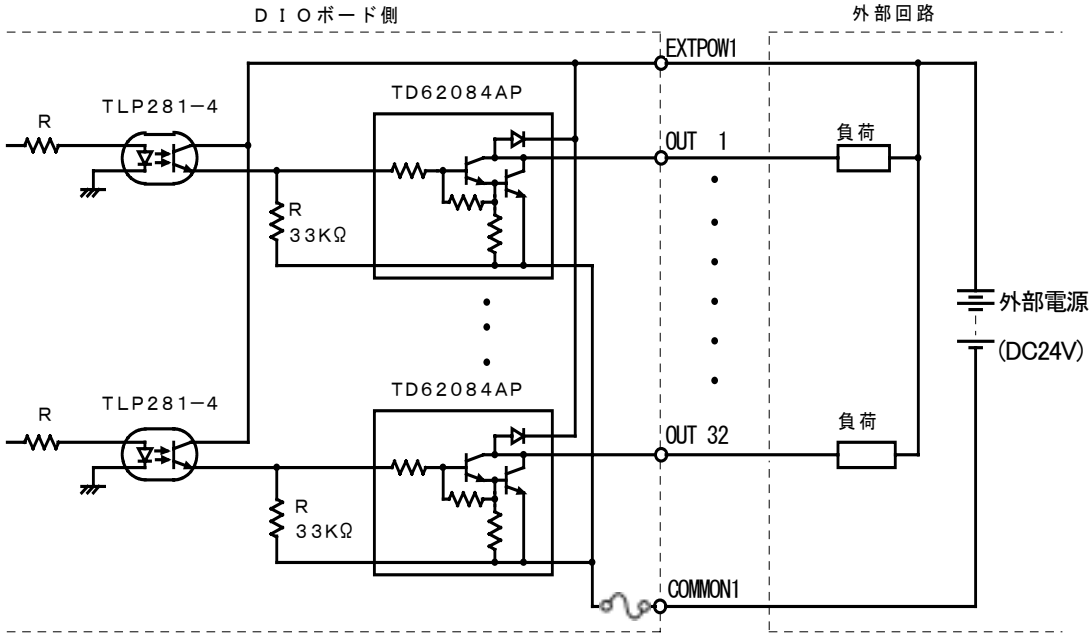


図２．８－１　出力回路接続例

２．８．２　入力回路接続例

本ボードの入力回路と外部回路がシンクタイプとの接続例を図２．８－２に示します.

（１）　外部回路がシンクタイプとの接続例

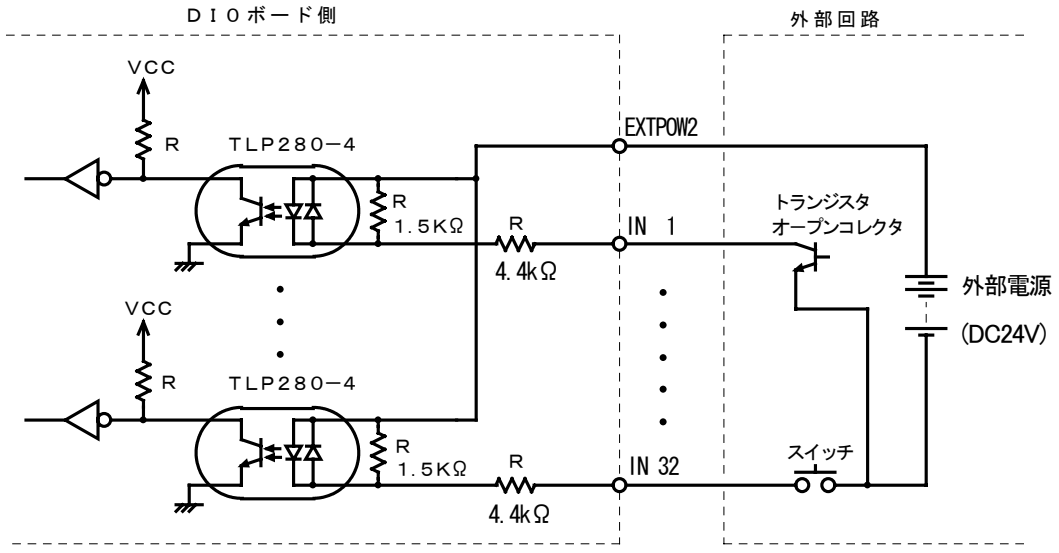


図２．８－２　シンクタイプとの接続

２．８．３　外部接続にあたっての注意事項

ランプやリレー等の誘導関係の負荷をコントロールする場合には、出力端子側で下記のよ うな対策を行って下さい.

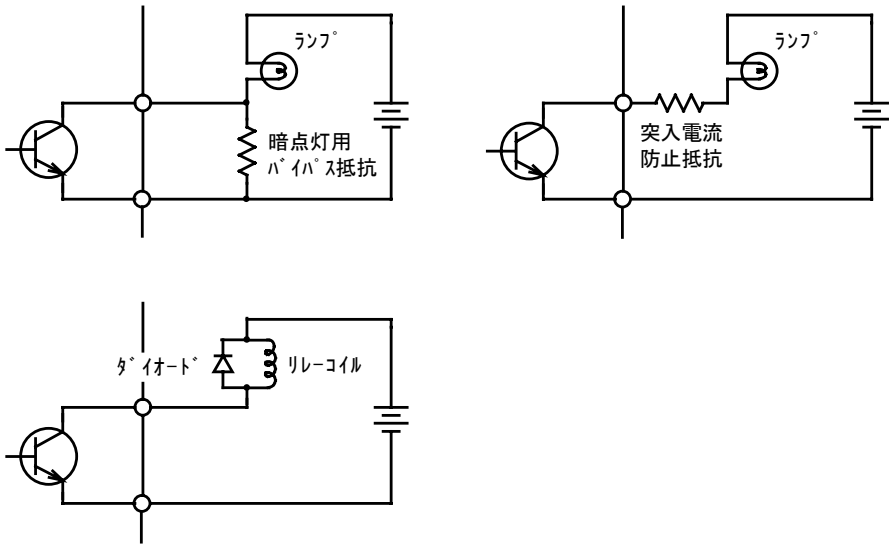


図２．８－３　外部接続における対策

３．ソフトウェア編

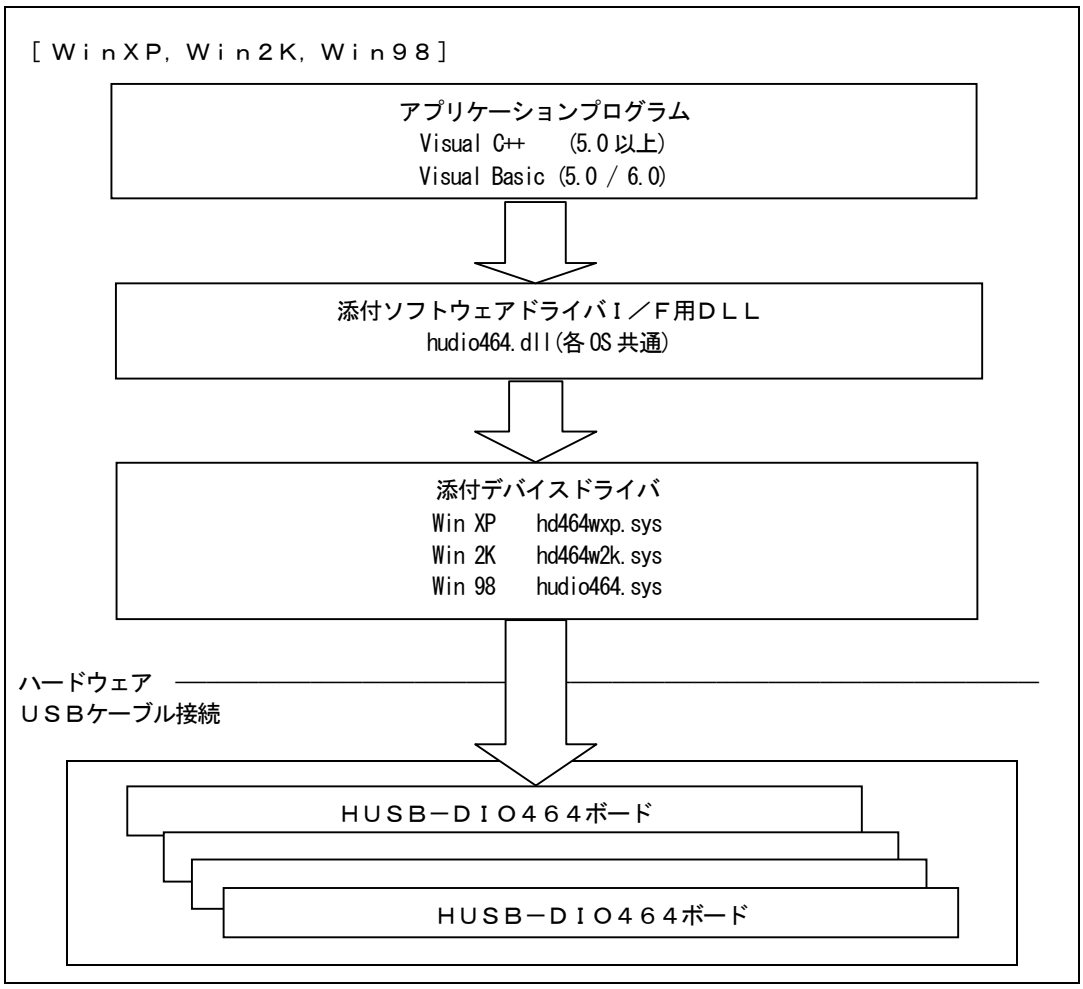
３．１　対応ＯＳ，ドライバ種別

（1）添付ソフトウェアの対応ＯＳ
　　ＷｉｎｄｏｗｓＸＰ Ｐｒｏｆｅｓｓｉｏｎａｌ，
　　ＷｉｎｄｏｗｓＸＰ Ｈｏｍｅ Ｅｄｉｔｉｏｎ　　（以下ＷｉｎＸＰとします），
　　Ｗｉｎｄｏｗｓ２０００ Ｐｒｏｆｅｓｓｉｏｎａｌ　（以下Ｗｉｎ２Ｋとします），
　　Ｗｉｎｄｏｗｓ９８ Ｓｅｃｏｎｄ Ｅｄｉｔｉｏｎ　（以下Ｗｉｎ９８とします）
　　です.

（2）Ｗｉｎｄｏｗｓ版デバイスドライバ　の種別は
　　ＷｉｎＸＰ　では　ｈｄ４６４ｗｘｐ．ｓｙｓ　です.
　　Ｗｉｎ２Ｋ　では　ｈｄ４６４ｗ２ｋ．ｓｙｓ　です.
　　Ｗｉｎ９８　では　ｈｕｄｉｏ４６４．ｓｙｓ　です.

（3）Ｗｉｎｄｏｗｓ版ドライバＩ/Ｆ用ＤＬＬ　として
　　ｈｕｄｉｏ４６４．ｄｌｌを上記ＯＳ共通で使用します.

３．２　添付ソフトウェアの構成



添付ソフトウェアのフォルダ構成は，（A:）¥Readme.txtを参照して下さい.

３．３　デバイスドライバのインストール

（１）ＷｉｎＸＰへのインストール

① Ｗｉｎｄｏｗｓ動作中を確認した後，ＤＩＯ４６４ボードの電源切状態でＵＳＢコネクタをパソコンのＵＳＢコネクタに挿入します．その後にＤＩＯ４６４ボードの電源を投入します．

② ＤＩＯ４６４ボードがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．

③ 添付ディスクをフロッピーディスクドライブに挿入します．

④ ソフトウェアを自動的にインストールする（推奨）を選択します．

⑤ HUSB-DIO464(WinXP)を指定してください．
後はシステムの指示に従ってインストールを完了させます．（図３．３－２を参照）

（２）Ｗｉｎ２Ｋへのインストール

① Ｗｉｎｄｏｗｓ動作中を確認した後，ＤＩＯ４６４ボードの電源切状態でＵＳＢコネクタをパソコンのＵＳＢコネクタに挿入します．その後にＤＩＯ４６４ボードの電源を投入します．

② ＤＩＯ４６４ボードがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．

③ システムがインストール元ディレクトリの指定を要求してきたら，添付ディスクをフロッピーディスクドライブに挿入します．

④ ◆″ 検索場所の指定 ″のチェックボックスを必ずチェックします．

⑤ Ａ：￥ＷＩＮ２Ｋ（Ａドライブである場合）を指定してください．
後はシステムの指示に従ってインストールを完了させます．（図３．３－３を参照）

（３）Ｗｉｎ９８へのインストール

① Ｗｉｎｄｏｗｓ動作中を確認した後，ＤＩＯ４６４ボードの電源切状態でＵＳＢコネクタをパソコンのＵＳＢコネクタに挿入します．その後にＤＩＯ４６４ボードの電源を投入します．

② ＤＩＯ４６４ボードがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．（初回のみ）

③ システムがインストール元ディレクトリの指定を要求してきたら，添付ディスクをフロッピーディスクドライブに挿入します．

④ ◆″ 検索場所の指定 ″のチェックボックスを必ずチェックします．

⑤ Ａ：￥ＷＩＮ９８（Ａドライブである場合）を指定してください．
後はシステムの指示に従ってインストールを完了させます．（図３．３－４を参照）

Windows X P
インストール結果

Windows2000
インストール結果

Windows98
インストール結果

図３．３－１　デバイスドライバインストール後の確認画面

図３．３－２　ＷｉｎＸＰ
のインストール手順

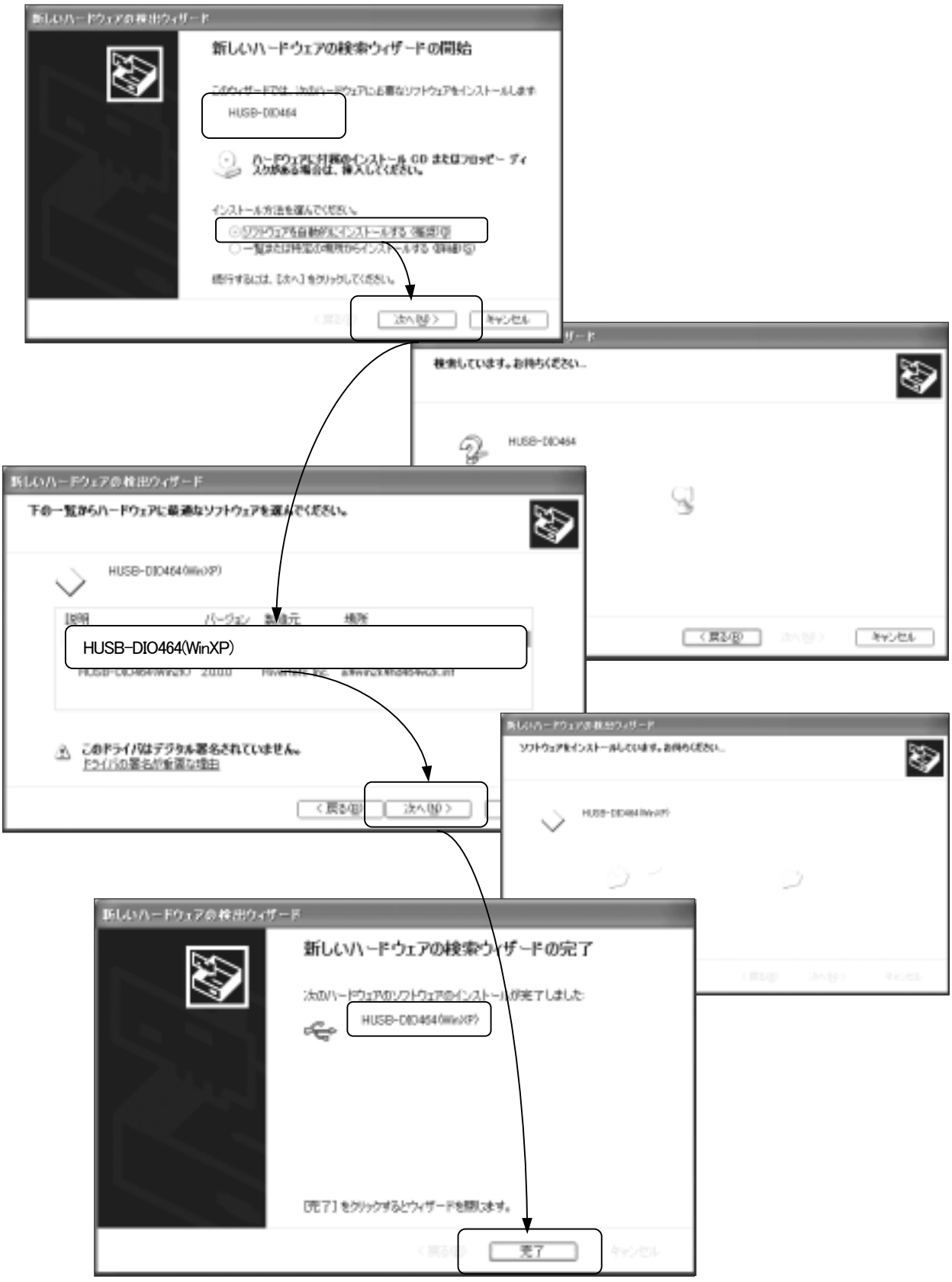


図３．３－３　Ｗｉｎ２Ｋ
のインストール手順

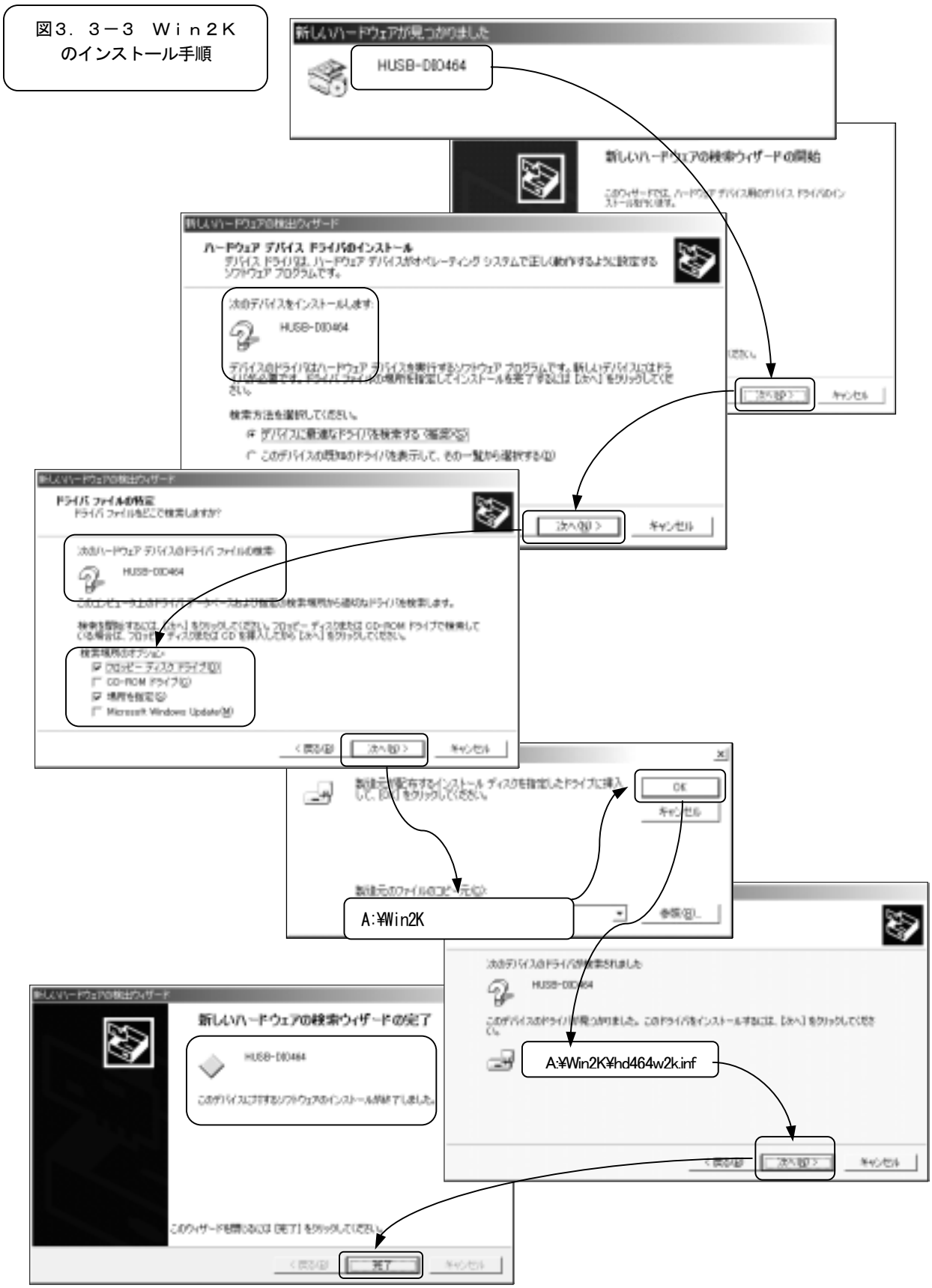
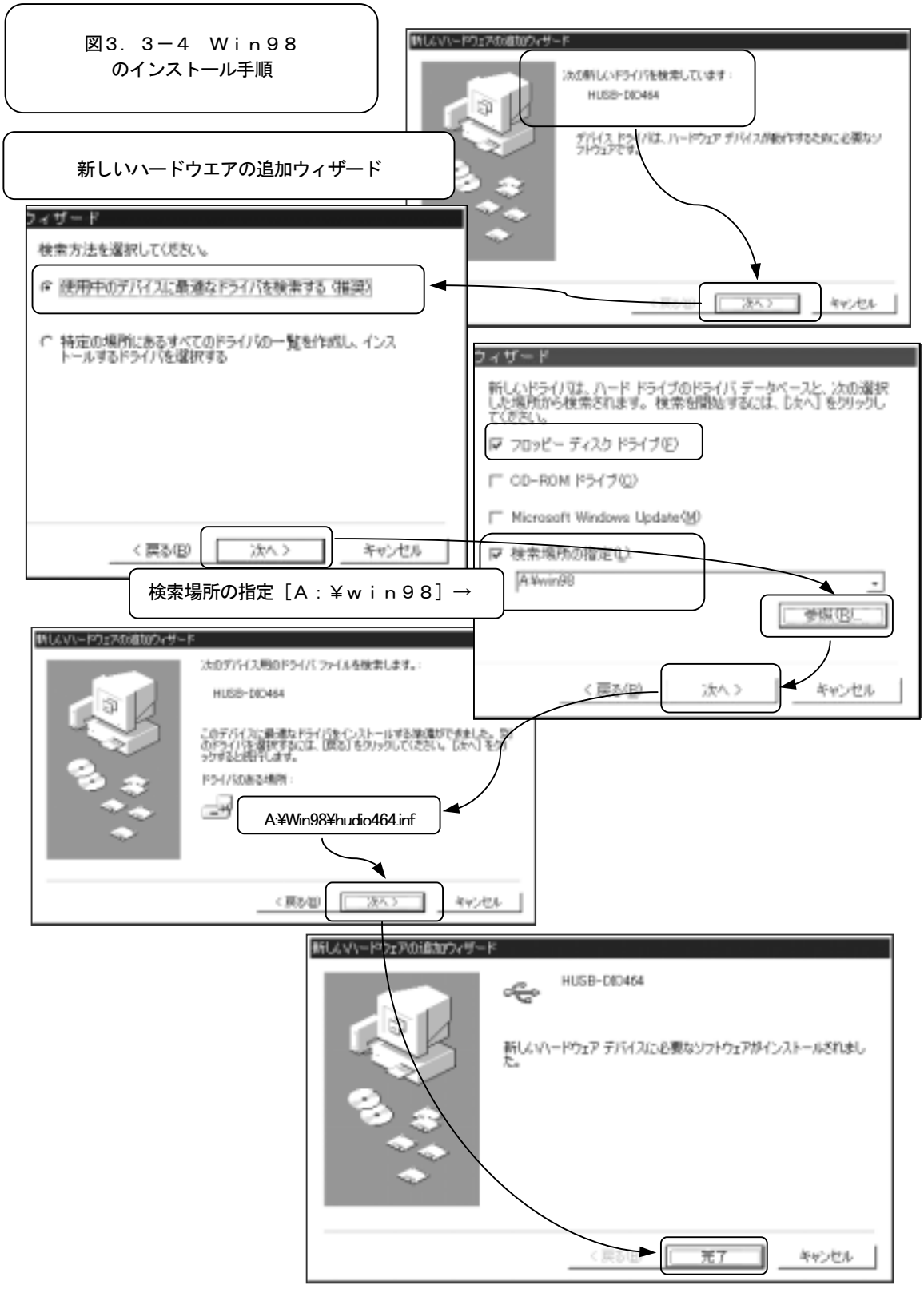


図３．３－４　Ｗｉｎ９８
のインストール手順



３．４　デバイスドライバの更新

ドライバ，ドライバI/Fライブラリのバージョンアップ等でデバイスドライバを更新する場合の方法は次の通りです.

３．４．１　ドライバの更新

更新（バージョンアップ）はドライバファイルの差し替えとなります.
更新を行う場合には，ＤＩＯボードをパソコンに接続した状態で行います.

（1）ＷｉｎｄｏｗｓＸＰ

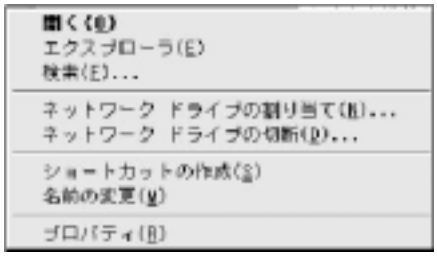
スタートメニューの「マイコンピュータ」アイコンを右クリックし，「プロパティ」をクリックします.
次に「ハードウェア」タブをクリックし，「デバイスマネージャ」をクリックします.
"Hivertec HUSB-DIO464 Driver" 左隣の "+" 記号をクリックして，一覧を展開します.
"HUSB-DIO464"を選択し，「ドライバの更新」をクリックします.
以降の手順は，図３．４－１　ＷｉｎＸＰドライバの更新を
参照して下さい.

（2）Ｗｉｎｄｏｗｓ２０００

デスクトップの「マイコンピュータ」アイコンを右クリックし，「プロパティ」をクリックします.
次に「ハードウェア」タブをクリックし，「デバイスマネージャ」をクリックします.
"Hivertec HUSB-DIO464 Driver" 左隣の "+" 記号をクリックして，一覧を展開します.
"HUSB-DIO464"を選択し，「プロパティ」をクリックします.
以降の手順は，図３．４－２　Ｗｉｎ２Ｋ ドライバの更新を参照して下さい.

（3）Ｗｉｎｄｏｗｓ９８

デスクトップの「マイコンピュータ」アイコンを右クリックし，「プロパティ」をクリックします.
次に「デバイスマネージャ」タブをクリックし，"Hivertec USB Driver" 左隣の "+" 記号をクリックして，一覧を展開します.
"HUSB-DIO464"を選択し，「プロパティ」をクリックします.
以降の手順は，図３．４－３　Ｗｉｎ９８ドライバの更新を参照して下さい.



３．４．２　ドライバＩ/Ｆライブラリの更新

Ｗｉｎｄｏｗｓのシステムディレクトリ（ＷｉｎＸＰならばC:￥Ｗｉｎｄｏｗｓ¥ｓｙｓｔｅｍ３２，Ｗｉｎ２ＫならばC:￥Ｗｉｎｎｔ¥ｓｙｓｔｅｍ３２，Ｗｉｎ９８ならばC:￥Ｗｉｎｄｏｗｓ¥ｓｙｓｔｅｍ（システムドライブがＣドライブの場合））にライブラリファイルを格納して使用している場合は上記のドライバの更新で更新されます.
もし，作業フォルダにライブラリファイルをコピーして使用されている場合は，作業フォルダに格納された同名のライブラリファイルを消去し，バージョンアップされたライブラリファイルを作業フォルダにコピーします.
（上書きコピーでも可能）

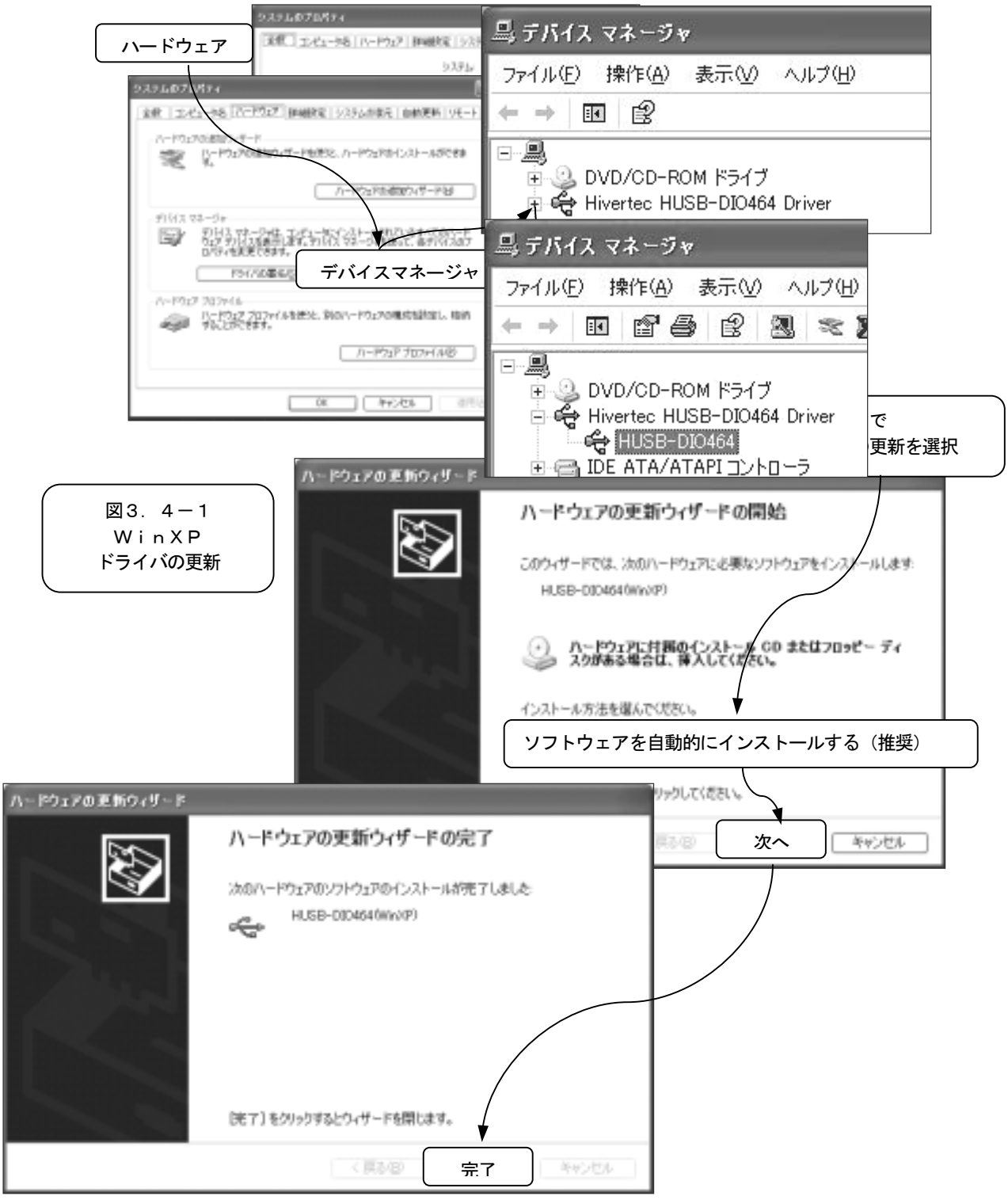


図３．４－１
ＷｉｎＸＰ
ドライバの更新

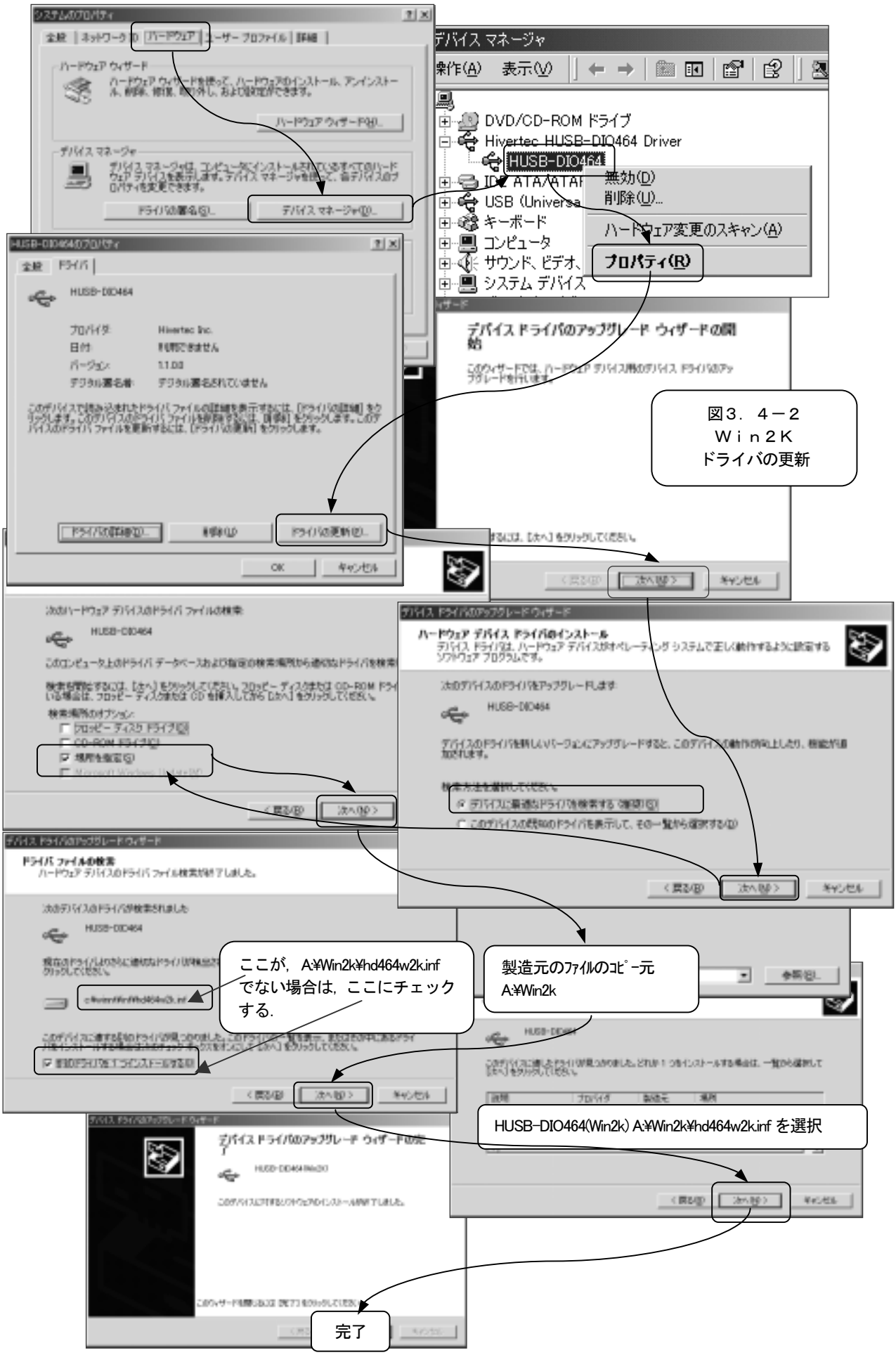
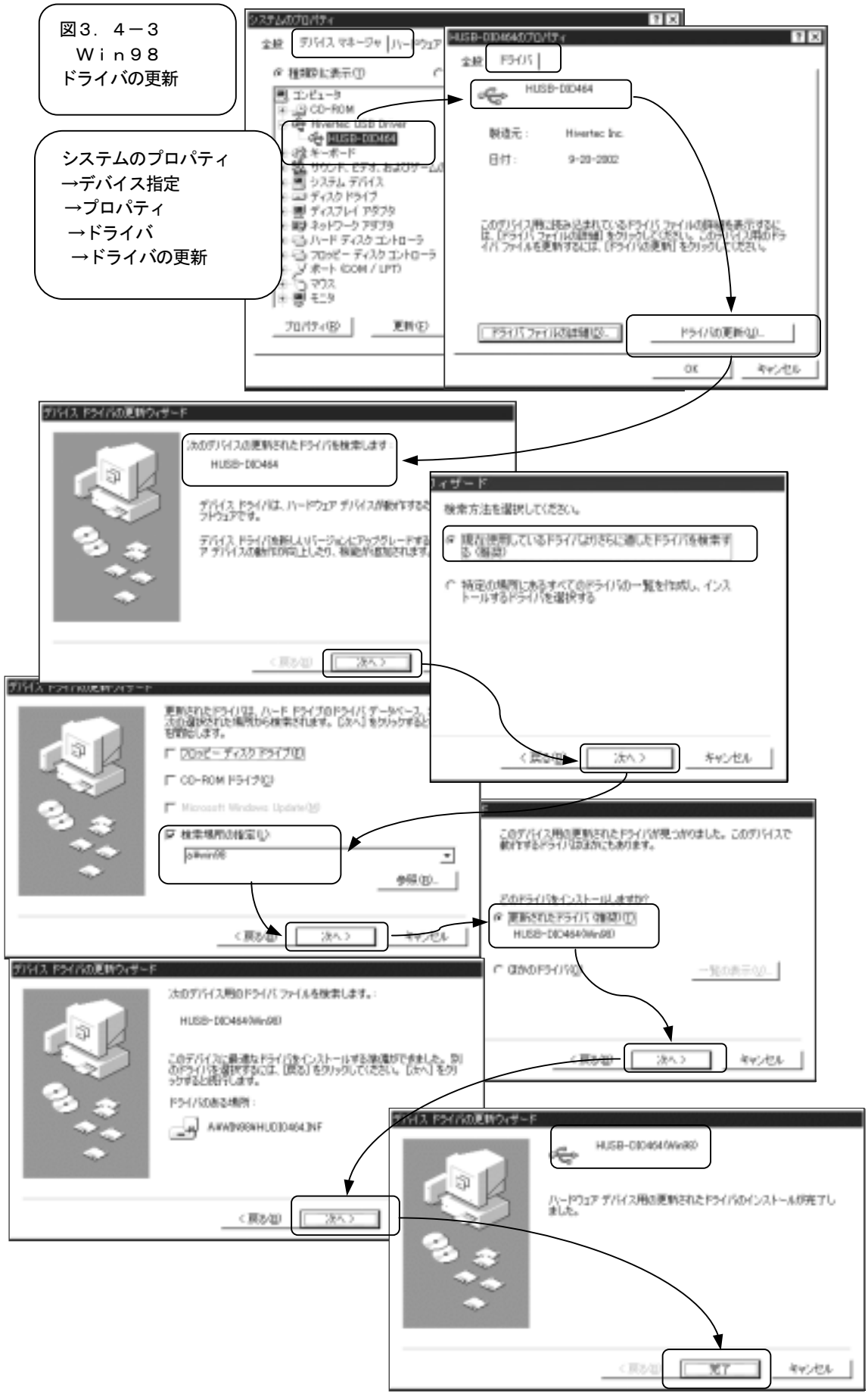


図３．４－２
Ｗｉｎ２Ｋ
ドライバの更新

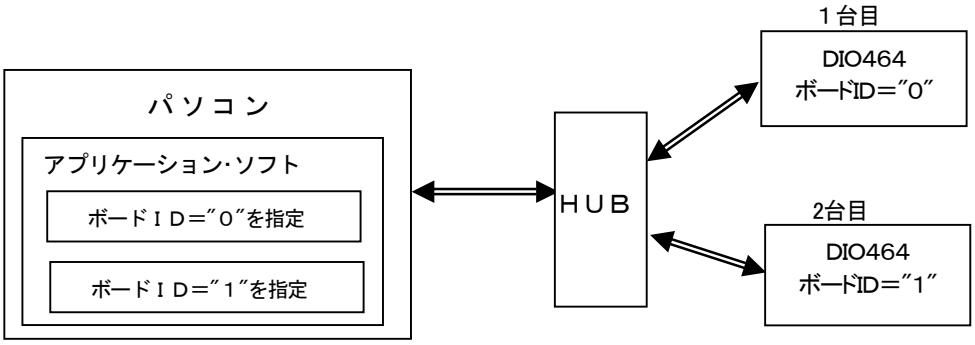
図３．４－３
Ｗｉｎ９８
ドライバの更新

システムのプロパティ
→デバイス指定
→プロパティ
→ドライバ
→ドライバの更新



３．５　ボードを複数枚使用する場合

ＤＩＯ４６４ボードを同一のコンピュータに複数枚接続し，それぞれのボードと外部の接続を１対１に対応させたい場合について説明します.



（1）ＨＵＳＢ－ＤＩＯ４６４コントローラの指定
ＵＳＢケーブルで接続された２台以上のＤＩＯボードは，パソコンによって内部で固定の番号が割り振られていますが，この番号をソフトウェアで利用できません.
そのため，同一型名のボードを識別するために，各ＤＩＯボードにそれぞれ固有な「ボードＩＤ Ｎｏ．」を設定し識別します.（ボードＩＤは０～１５）

（2）デバイスＩＤの確認方法
添付ソフトウェアフロッピーディスクのＶＣサンプル「¥sample¥Vc¥Release¥sud46400.exe」を実行し，画面のボードＩＤを確認します.

（3）ボードＩＤの設定方法
ボードＩＤの設定については　３．３　ボード上の設定「図３．３－２　ボードＩＤの設定」を参照して下さい.

３．６　ＵＳＢボードの認識

ＵＳＢボードの認識・接続・切り離し

	項　　目	備　　考
1	ＤＩＯボード（デバイス）の認識	アプリケーションプログラム（ドライバＩ/Ｆ用ＤＬＬ）起動時に行われます
2	起動時以降に接続されたデバイス	認識されません
3	起動時以降に切り離されたデバイス	制御不能となります
4	アプリケーション実行中に任意のＵＳＢ接続	アプリケーションのＵＳＢ通信に影響を与えます（通信停止状態が発生）
5	ＤＩＯボード（デバイス）の着脱	接続では５秒以上経過してから使用（通信）して下さい 切り離し後の再接続は，５秒以上経過してから行います

３．７　ボードアクセス方法

複数のＤＩＯ４６４ボードを制御することができます.
ある１つのＤＩＯ４６４ボードにアクセスするためには,まずこのデバイスをオープンして,アクセスするための足がかりとなるデバイスハンドル値を取得する必要があります.
デバイスをオープンするためには,どのようなハードウェアリソースを持つデバイスをオープンするかという情報が必要となります.

３．７．１　ボード（デバイス）認識用のデータ構造体

ボード認識のために次に示す HUSBDEVINF 型構造体が用意されています.

[C言語 : Visual C/C++]

```
typedef struct _HUSBDEVINF {
  WORD        BrdID;          // ボードID        ( 0～15 )
  WORD        EPcnt;          // エンドポイント数      (>=2)
  WORD        EPspc[8];       // エンドポイント仕様    (0x81,0x02,0x00,0x00)
  WORD        EPsiz[8];       // エンドポイント・バイト数(32,32, 0, 0)
  WORD        EPopt[8];       // エンドポイント・オプション( 予約 )
  HANDLE      EPhdl[8];       // エンドポイント・ハンドル(パイプハンドル)
} HUSBDEVINF, * PHUSBDEVINF;
```

[Visual Basic]

```
Type spc_mat
    dt(0 To 7)      As Integer
End Type
Type hdl_mat
    dt(0 To 7)      As Long
End Type
Public Type HUSBDEVINF
    BrdID       As Integer      ' ボードID        ( 0～15 )
    EPcnt       As Integer      ' エンドポイント数      (>=2)
    EPspc       As spc_mat      ' エンドポイント仕様    (0x81,0x02,0x00,0x00)
    EPsiz       As spc_mat      ' エンドポイント・バイト数(32,32, 0, 0)
    EPopt       As spc_mat      ' エンドポイント・オプション( 予約 )
    EPhdl       As hdl_mat      ' エンドポイント・ハンドル(パイプハンドル)
End Type
```

３．７．２　ドライバ関数の使用

（1）Ｖｉｓｕａｌ C++（5．0 以上）によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
■プロジェクト追加ファイル
◇ドライバ　関数用・・**hudio464.lib**　・・　ドライバ関数インポートライブラリ
■インクルードファイル
◇ドライバ　関数用・・**hudio464.h**　・・　ドライバ関数結合用ヘッダーファイル

（2）Ｖｉｓｕａｌ Ｂａｓｉｃ（5．0／6．0）によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
◇ドライバ　関数用・・**hudio464.bas**　・・　ドライバＩ／Ｆ用ＤＬＬ関数定義標準モジュールファイル
このファイルに外部関数宣言（Declare），及びユーザー定義型宣言が記述されています.

３．７．３　C++アプリケーションでの使用

ドライバ関数は**「C言語」**で作成されています．これらの関数をC++アプリケーションで使用できるように，ドライバ関数のヘッダーファイル内で以下のように「Ｃ言語」として明示的な定義をしています.

```
//―――――――――――――――――――――――――――――――――
//          関数プロトタイプ宣言
//―――――――――――――――――――――――――――――――――
#ifdef __cplusplus
extern "C"
{
#endif

          個々の関数のプロトタイプ宣言

#ifdef __cplusplus
}
#endif
```

（1）　#ifdef __cplusplus・・Ｖｉｓｕａｌ C++ 用の定義です
#endif　　C++コーディングでは明示的にライブラリ関数が”Ｃモジュール”として解釈されます.

（2）　extern "C"　　・・　Ｃモジュールで定義されている関数を表します.

３．７．４　関数一覧

No	関数名称	機　能
1	dio464_GetDeviceCount ()	ボード枚数の取得
2	dio464_GetDeviceInfo ()	デバイス情報の取得
3	dio464_OpenDevice ()	デバイスオープン
4	dio464_CloseDevice ()	デバイスクローズ
5	dio464_rInB ()	入力ポートからの１バイトデータ読込
	dio464_rOutB ()	出力ポートからの１バイトデータ読込
	dio464_wOutB ()	出力ポートへの　１バイトデータ書込
	dio464_rInW ()	入力ポートからの２バイトデータ読込
	dio464_rOutW ()	出力ポートからの２バイトデータ読込
	dio464_wOutW ()	出力ポートへの　２バイトデータ書込
6	dio464_rInDW ()	入力ポートからの４バイトデータ読込
	dio464_rOutDW ()	出力ポートからの４バイトデータ読込
	dio464_wOutDW ()	出力ポートへの　４バイトデータ書込
7	dio464_InOutDW ()	出力ポートへの４バイトデータ書込と 入力ポートからの４バイトデータ読込
8	dio464_SetFilter ()	入力ポートのフィルターの設定

表３．７－１　関数一覧

３．７．５　ＵＳＢボードとの通信時間

ＨＵＳＢ－ＤＩＯボードは［ＵＳＢ １.１ 規格 ：バルク通信 ］より，ボードとの通信に多少の時間を要します．
ドライバ関数の概略所要時間を次の表に示します．（個々のパソコンで変化します．）

No	関数名称	通信時間（msec）
1	dio464_GetDeviceCount ()	――
2	dio464_GetDeviceInfo ()	――
3	dio464_OpenDevice ()	――
4	dio464_CloseDevice ()	――
5	dio464_rInB ()	4
	dio464_rOutB ()	4
	dio464_wOutB ()	2
	dio464_rInW ()	4
	dio464_rOutW ()	4
	dio464_wOutW ()	2
6	dio464_rInDW ()	4
	dio464_rOutDW ()	4
	dio464_wOutDW ()	2
7	dio464_InOutDW ()	4
8	dio464_SetFilter ()	2

表３．７－２　関数通信時間

（注）本表の所要時間値は，ドライバ関数を連続して使用する場合であり，関数単独使用の場合は″1msec″減じた値です．

３．７．６　ボードアクセスの準備手順

（１）使用する全ボードのデバイス情報の取得

"HUSBDEVINF"型構造体エリア（の配列）内に，全ＤＩＯ４６４のデバイス情報をまず取得します.

◆ dio464_GetDeviceCount()・・・ボード枚数の確認

◆ dio464_GetDeviceInfo()　・・・全ボードのデバイス情報を取得

（２）ボード毎にデバイスオープン

ある１つのＤＩＯボードのデバイス情報をデバイスオープン関数に渡します.

この結果そのＤＩＯボードがオープンされ，デバイスオープン関数はこのボードにアクセスするための
デバイスハンドル値を返してきます.

ボード枚数が２枚以上の場合には，個々のボード毎にこの処理を行います.

特定のボードを選択する場合には，ボードＩＤ値をチェックして下さい.

◆ dio464_OpenDevice()　　・・・ボードのオープン処理

以降は，この「デバイスハンドル」を使用し，そのボードにアクセスすることができるようになります.

（３）全ての処理が終了してアプリケーションを終了する場合には，オープンしたデバイスの「クローズ処理」を
行って下さい.

◆ dio464_CloseDevice()　　・・・ボードのクローズ処理（１枚分）

*******　デバイス判別の特例　*******

デバイス情報取得時に，デバイスが認識されない，ボード設定のボードＩＤが見つからない場合が
あります.

■次の場合にはＤＩＯデバイスとしては認めていません.　（デバイス情報なし）

◆ベンダＩＤ・プロダクトＩＤが一致しても，デバイスからの情報取得に失敗.

◆デバイスのエンドポイント数が不足.　　（入出力不可）

◆各エンドポイントの仕様が一致しない.　（入出力不可）

■上記の条件を満足しても，次の場合にはボードＩＤ＝９９として，ドライバは処理しています.

◆ＤＩＯボードの持つ一部の機能が認識できない.

このケースでは，デバイス情報の取得はできますが，ＤＩＯボードとしての正常な動作は
保証しかねます.

３．８　ドライバＩ/Ｆ用ＤＬＬ関数詳細

Ｎｏ	1	dio464_GetDeviceCount()	ボード枚数の取得
機　能		現在パソコンに装着されているＤＩＯボードの枚数を取得します.	
VC++	書　式	DWORD WINAPI dio464_GetDeviceCount(DWORD*　count);	
	引　数	DWORD*　　count・・取得したＤＩＯボード枚数の格納エリア	
	呼出例	DWORD*　count;　　//ＤＩＯボードの枚数 DWORD　　ret;　　//関数の戻り値 ret = dio464_GetDeviceCount(&count);	
ＶＢ	書　式	Declare Function dio464_GetDeviceCount Lib "hudio464.dll" (ByRef count As Long) As Long	
	引　数	ByRef count　As Long　　・・取得したＤＩＯボード枚数の格納エリア	
	呼出例	Dim count　As Long　　'ＤＩＯボードの枚数 Dim ret　　As Long　　'関数の戻り値 ret = dio464_GetDeviceCount(count)	
備　　考			

Ｎｏ	2	dio464_GetDeviceInfo()	デバイス情報の取得
機　能		現在パソコンに装着されている指定枚数ＤＩＯボードのデバイス情報を取得します. この結果、デバイス情報構造体の配列にデバイス情報が格納されます. このデバイス情報は、デバイスオープン時に利用します.	
VC++	書　式	DWORD WINAPI dio464_GetDeviceInfo(DWORD* count, HUSBDEVINF* Husbinf);	
	引　数	DWORD*　　count　　・・ＤＩＯボードの枚数（デバイス情報取得ボード枚数） HUSBDEVINF*　Husbinf・・取得するデバイス情報がセットされるエリア先頭アドレス	
	呼出例	DWORD　　ret;　　//関数の戻り値 DWORD　　count = 2;　　//最大枚数は２ HUSBDEVINF　Husbinf[2];　　//２枚のＤＩＯボードのデバイス情報がセットされるべきエリア ret = dio464_GetDeviceInfo(&count, &Husbinf[0]);	
ＶＢ	書　式	Declare Function dio464_GetDeviceInfo Lib "hudio464.dll" (ByRef count As Long,Husbinf As HUSBDEVINF) As Long	
	引　数	ByRef count　As Long　　・・ＤＩＯボードの枚数（デバイス情報取得ボード枚数） Husbinf　As HUSBDEVINF・・取得するデバイス情報がセットされるエリア先頭アドレス	
	呼出例	Dim ret　　As Long　　'関数の戻り値 Dim count　　As Long　　'枚数 Dim Husbinf(2)　As HUSBDEVINF　　'デバイス情報のエリア count = 2　　'最大枚数は２枚 ret = dio464_GetDeviceInfo(count, Husbinf(0))	
備　　考			

Ｎｏ	３	dio464_OpenDevice()	デバイスのオープン
機　能		渡したデバイス情報を持つＤＩＯボードをオープンし、他と識別するためのデバイスハンドルを取得します. 以降このデバイスハンドルは、このＤＩＯボードにアクセスするためのハンドルとなります. また, この時出力ポートはすべて”０”になります.	
VC++	書　式	DWORD WINAPI dio464_OpenDevice(DWORD * hDev, HUSBDEVINF* Husblnf);	
	引　数	DWORD hDev; ・・取得するデバイスハンドルの格納エリア HUSBDEVINF* Husblnf; ・・オープンするボードのデバイス情報格納アドレス	

	呼出例	DWORD ret; //関数の戻り値
		DWORD hDev[2]; //デバイスハンドル取得エリア
		ret = dio464_OpenDevice(&hDev[0], &Husblnf[0]); //１番目のデバイス情報
		ret = dio464_OpenDevice(&hDev[1], &Husblnf[1]); //２番目のデバイス情報	

ＶＢ	書　式	Declare Function dio464_OpenDevice Lib "hudio464.dll" (ByRef hDev As Long, Husblnf As HUSBDEVINF) As Long	
	引　数	ByRef hDev As Long ・・取得するデバイスハンドルの格納エリア Husblnf As HUSBDEVINF ・・オープンするボードのデバイス情報格納アドレス	
	呼出例	Dim ret As Long '関数の戻り値 Dim hDev(2) As Long 'デバイスハンドル取得エリア ret = dio464_OpenDevice(hDev(0), Husblnf(0)) '１番目のデバイス情報 ret = dio464_OpenDevice(hDev(1), Husblnf(1)) '２番目のデバイス情報	
備　考			

Ｎｏ	４	dio464_CloseDevice()	デバイスのクローズ
機　能		渡したデバイスハンドルを持つＤＩＯボードをクローズします. 以降このデバイスハンドルは、無効となり, このＤＩＯボードにアクセスはできません. また, 出力ポートの状態は最終出力状態が保持されます.	
VC++	書　式	DWORD WINAPI dio464_CloseDevice(DWORD hDev);	
	引　数	DWORD hDev ・・クローズするボードのデバイスハンドル	

	呼出例	DWORD ret; //関数の戻り値
		ret = dio464_CloseDevice(hDev);	

ＶＢ	書　式	Declare Function dio464_CloseDevice Lib "hudio464.dll" (ByVal hDev As Long) As Long	
	引　数	ByVal hDev As Long ・・クローズするボードのデバイスハンドル	
	呼出例	Dim ret As Long '関数の戻り値 ret = dio464_CloseDevice(hDev)	
備　考			

<table>
<tr><td rowspan="6">Ｎｏ</td><td rowspan="6">５</td><td>dio464_rInB()</td><td>入力ポートからの１バイトデータ読込</td></tr>
<tr><td>dio464_rOutB()</td><td>出力ポートからの１バイトデータ読込</td></tr>
<tr><td>dio464_wOutB()</td><td>出力ポートへの１バイトデータ書込</td></tr>
<tr><td>dio464_rInW()</td><td>入力ポートからの２バイトデータ読込</td></tr>
<tr><td>dio464_rOutW()</td><td>出力ポートからの２バイトデータ読込</td></tr>
<tr><td>dio464_wOutW()</td><td>出力ポートへの２バイトデータ書込</td></tr>
<tr><td colspan="2">機　能</td><td colspan="2">デバイスハンドルで指定されたＤＩＯボードの指定された入力ポートまたは出力ポートから
入力ポートからの１バイトデータ読込・・１バイトデータを読込み指定したエリアに格納します。
出力ポートからの１バイトデータ読込・・１バイトデータを読込み指定したエリアに格納します。
出力ポートへの１バイトデータ書込　・・１バイトデータを書込ます。
入力ポートからの２バイトデータ読込・・２バイトデータを読込み指定したエリアに格納します。
出力ポートからの２バイトデータ読込・・２バイトデータを読込み指定したエリアに格納します。
出力ポートへの２バイトデータ書込　・・２バイトデータを書込ます。
出力ポートの読込により最終出力状態の確認ができます。</td></tr>
<tr><td rowspan="3">VC++</td><td>書　式</td><td colspan="2">DWORD WINAPI dio464_rInB (DWORD hDev, WORD port, WORD* wrdt);
DWORD WINAPI dio464_rOutB(DWORD hDev, WORD port, WORD* wrdt);
DWORD WINAPI dio464_wOutB(DWORD hDev, WORD port, WORD wwdt);
DWORD WINAPI dio464_rInW (DWORD hDev, WORD port, WORD* wrdt);
DWORD WINAPI dio464_rOutW(DWORD hDev, WORD port, WORD* wrdt);
DWORD WINAPI dio464_wOutW(DWORD hDev, WORD port, WORD wwdt);</td></tr>
<tr><td>引　数</td><td colspan="2">DWORD hDev ・・ 対象デバイスのデバイスハンドル
WORD port ・・ ポート指定(0:ﾎﾟｰﾄ1, 1:ﾎﾟｰﾄ2, 2:ﾎﾟｰﾄ3, 3:ﾎﾟｰﾄ4)
WORD* wrdt ・・ 読込んだデータを格納する２バイトエリアのアドレス
WORD wwdt ・・ 書込む２バイトデータ</td></tr>
<tr><td>呼出例</td><td colspan="2">DWORD ret; //関数の戻り値
WORD dt; //データ格納エリア
ret = dio464_rInB(hDev, 0, &dt); //入力ポート１の読込</td></tr>
<tr><td rowspan="3">ＶＢ</td><td>書　式</td><td colspan="2">Declare Function dio464_rInB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByRef wrdt As Integer) As Long
Declare Function dio464_rOutB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByRef wrdt As Integer) As Long
Declare Function dio464_wOutB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByVal wwdt As Integer) As Long
Declare Function dio464_rInW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByRef wrdt As Integer) As Long
Declare Function dio464_rOutW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByRef wrdt As Integer) As Long
Declare Function dio464_wOutW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _
ByVal wwdt As Integer) As Long</td></tr>
<tr><td>引　数</td><td colspan="2">ByVal hDev As Long ・・ 対象デバイスのデバイスハンドル
ByVal port As Integer ・・ ポート指定(0:ﾎﾟｰﾄ1, 1:ﾎﾟｰﾄ2, 2:ﾎﾟｰﾄ3, 3:ﾎﾟｰﾄ4)
ByRef wrdt As Integer ・・ 読込んだデータを格納する２バイトエリアのアドレス
ByVal wwdt As Integer ・・ 書込む２バイトデータ</td></tr>
<tr><td>呼出例</td><td colspan="2">Dim ret As Long '関数の戻り値
Dim dt As Integer 'データ格納エリア
ret = dio464_ rInB(hDev, 0, dt); '入力ポート１の読込</td></tr>
</table>

備　　考	１バイトの入出力関数の場合 読込データは　上位バイト：0x00 　　下位バイト：読込データ 書込データは　上位バイト：無視 　　下位バイト：書込データ

Ｎｏ	６	dio464_rInDW()	入力ポートからの４バイトデータ読込
		dio464_rOutDW()	出力ポートからの４バイトデータ読込
		dio464_wOutDW()	出力ポートへの４バイトデータ書込
機　能		デバイスハンドルで指定されたＤＩＯボードの入力ポートまたは出力ポートから 入力ポートからの４バイトデータ読込・・４バイトデータを読込み指定したエリアに格納します。 出力ポートからの４バイトデータ読込・・４バイトデータを読込み指定したエリアに格納します。 出力ポートへの４バイトデータ書込　・・４バイトデータを書込ます。 出力ポートの読込により最終出力状態の確認ができます。	
VC++	書　式	DWORD WINAPI dio464_rInDW (DWORD hDev, DWORD* drdt); DWORD WINAPI dio464_rOutDW(DWORD hDev, DWORD* drdt); DWORD WINAPI dio464_wOutDW(DWORD hDev, DWORD dwdt);	
	引　数	DWORD hDev　・・ 対象デバイスのデバイスハンドル DWORD* drdt　・・ 読込んだデータを格納する４バイトエリアのアドレス DWORD dwdt　・・ 書込む４バイトデータ	
	呼出例	DWORD ret;　　//関数の戻り値 DWORD dt;　　//データ格納エリア ret = dio464_rInDW(hDev, &dt);　//入力ポート読込	
ＶＢ	書　式	Declare Function dio464_rInDW Lib "hudio464.dll" (ByVal hDev As Long, ByRef drdt As Long) As Long Declare Function dio464_rOutDW Lib "hudio464.dll" (ByVal hDev As Long, ByRef drdt As Long) As Long Declare Function dio464_wOutDW Lib "hudio464.dll" (ByVal hDev As Long, ByVal dwdt As Long) As Long	
	引　数	ByVal hDev　As Long　・・ 対象デバイスのデバイスハンドル ByRef drdt　As Long　・・ 読込んだデータを格納する４バイトエリアのアドレス ByVal dwdt　As Long　・・ 書込む４バイトデータ	
	呼出例	Dim ret　As Long　　'関数の戻り値 Dim dt　As Long　　'データ格納エリア ret = dio464_ rInDW(hDev, dt);　'入力ポート読込	
備　　考			

No	7	dio464_InOutDW()	ＤＩＯの出力ポートへ４バイトデータ書込と ＤＩＯ入力ポートからの４バイトデータ読込
機　能		デバイスハンドルで指定されたＤＩＯの出力ポートへ４バイトデータを書込ます。 また出力直前の入力ポートから４バイトデータを読込み、指定したエリアに格納します。	
VC++	書　式	DWORD WINAPI dio464_InOutDW (DWORD hDev, DWORD* drdt, DWORD dwdt);	
	引　数	DWORD hDev ・・ 対象デバイスのデバイスハンドル DWORD* drdt ・・ 読込んだデータを格納する４バイトエリアのアドレス DWORD dwdt ・・ 書込む４バイトデータ	

	呼出例	DWORD ret; //関数の戻り値
		DWORD rdt; //データ格納エリア
		ret = dio464_InOutDW(hDev, &rdt, 0x55555555); //55555555h のデータ出力と入力ポート読込	

ＶＢ	書　式	Declare Function dio464_InOut Lib "hudio464.dll" (ByVal hDev As Long, _ ByRef drdt As Long _ ByVal dwdt As Long) As Long	
	引　数	ByVal hDev As Long ・・ 対象デバイスのデバイスハンドル ByRef drdt As Long ・・ 読込んだデータを格納する４バイトエリアのアドレス ByVal dwdt As Long ・・ 書込む４バイトデータ	
	呼出例	Dim ret As Long '関数の戻り値 Dim dt As Long 'データ格納エリア ret = dio464_ InOutDW(hDev, dt, &H55555555) '55555555h のデータ出力と入力ポート読込	
備　考			

No	8	dio464_SetFilter()	入力ポートのフィルターの設定
機　能		デバイスハンドルで指定されたＤＩＯの指定された入力ポートのフィルターを設定します	
VC++	書　式	DWORD WINAPI dio464_SetFilter (DWORD hDev, WORD port, DWORD wtm);	
	引　数	DWORD hDev ・・ 対象デバイスのデバイスハンドル WORD port ・・ ポート指定 (0x1 : ﾎﾟｰﾄ1, 0x2 : ﾎﾟｰﾄ2, 0x4 : ﾎﾟｰﾄ3, 0x8 : ﾎﾟｰﾄ4) DWORD wtm ・・ フィルター時間 (wtm × 0.1msec)	

	呼出例	DWORD ret; //関数の戻り値
		ret = dio464_ SetFilter(hDev, 0x1, 100); //ポート１に 10msec のフィルタ設定	

ＶＢ	書　式	Declare Function dio464_SetFilter Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer _ ByVal wtm As Integer) As Long	
	引　数	ByVal hDev As Long ・・ 対象デバイスのデバイスハンドル ByVal port As Integer ・・ ポート指定 (&H1 : ﾎﾟｰﾄ1, &H2 : ﾎﾟｰﾄ2, &H4 : ﾎﾟｰﾄ3, &H8 : ﾎﾟｰﾄ4) ByVal wtm As Integer ・・ サンプリング時間(wtm × 0.1msec)	
	呼出例	Dim ret As Long '関数の戻り値 ret = dio464_SetFilter(hDev, &H1, 100) 'ポート１に 10msec のフィルター設定	
備　考		ポート指定が複数の場合はＯＲしたデータを与えます. wtm の設定範囲は 1 ～ 65535 (ffffh) になります. 従って, サンプリング時間の設定範囲は 0.1msec ～ 6553.5msec になります. また, 入力に変化が発生してから, ボード内に取り込まれた入力状態が確定するまでに, 最小 wtm ～ 最大 ２ × wtm の時間を要します.	

３．９　関数の戻り値

ドライバの関数を使用する時，関数の戻り値が異常値（’０’以外）であった場合には，異常内容に対応した処理を行います.
通常，この異常が発生した場合にはアプリケーションプログラムの続行は困難であり，プログラム内容の再検討が必要となります.

Ｎｏ	戻り値			異 常 内 容 と 確 認 項 目
	記号表記	１６進数表記		
		VC++	ＶＢ	
１	NO_ERROR	0x000	&H0	正　　常 異常は発生していません
２	NOT_FOUND	0x001	&H1	デバイスドライバが存在しない ◎デバイスドライバがインストールされていない ◎デバイスドライバが所定のフォルダに格納されていない
３	ALREADY_OPENED	0x002	&H2	既にオープン済のデバイスをオープン ◎オープン済みデバイスに更にオープン指令 ◇オープンしたデバイスはクローズするまで使用 （多重のオープンは禁止） ◎ボード２枚以上使用する場合,オープンするデバイス情報の更新を確認します.
４	NOT_MEMORY	0x004	&H4	デバイス情報格納メモリが不足 ◎アプリケーション用のメモリ不足 ◇パソコン主記憶メモリの不足 ◎システムリソース（ＯＳ用メモリ）の不足 ◇多数のアプリケーション起動 ◇１度に多数のウインドウを開いた
５	INVALID_HANDLE	0x008	&H8	無効なデバイスハンドルを指定 ◎デバイスオープンで得られた"デバイスハンドル"の不使用 ◎このデバイスは既にクローズされている
６	NOT_READY	0x010	&H10	デバイスの入出力ポートが使用できない ◎デバイス（ボード）内部の入出力ポートがない システムとの不整合等が考えられますので、弊社サポートまでお問い合わせください
７	ILLEGAL_DEVICE	0x020	&H20	ボード固有情報が不正 ◎デバイス情報は正常ですが,ボード機能が不一致です
８	ILLEGAL_PARAM	0x100	&H100	関数の引数の値が異常 ◇引数の設定値を確認（マニュアル照合）

３．１０　サンプルプログラム

ドライバ関数の使用方法を解説する目的のサンプルプログラムを添付しています.
サンプルプログラムは次の２種類があり，ほぼ同一の画面表示と操作になっています.
以降のサンプルプログラム説明では，①の「Ｃコーディング」を用います.

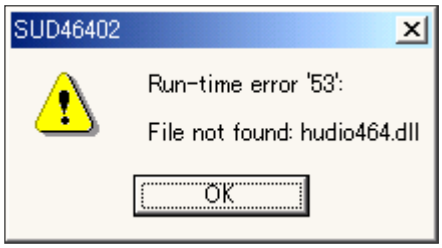
① Ｖisual Ｃ++ (5.0)・・ Ｃコーディング 【 sud46400.exe 】
② Ｖisual Ｂasic(5.0) 【 sud46402.exe 】

サンプルプログラムを使用する場合は，お客様のハードディスクにコピーして使用します.
個々のサンプル実行ファイル（sud46400.exe，sud46402.exe）は”マウスのダブルクリック”操作を行う事で実行できます.

≪　ご注意　≫

◆Visual C++ サンプルは開発ツールとして Visual C++ 5.0 以上がインストールされている必要があります.
◆Visual Basic サンプルは開発ツールとして Visual Basic 5.0 がインストールされている必要があります.
◆ＤＩＯ４６４ボードを２枚以上で使用する場合，ボードＩＤ は重複しないようにして下さい.
◆実行開始時に次頁のエラーメッセージが表示される場合には，プログラムは動作しません.

【　エラーメッセージの表示　】



※ DLL がインストールされていない.
DLL をインストールして下さい.

※　ＤＩＯ４６４ボードが装着されていない.
または，システムが認識していない.
※　デバイスドライバがインストールされていない.
ボードを装着してください．またはデバイスドライバをインストールしてください.

３．１０．１　サンプルプログラムの操作

サンプルプログラムが正常に起動されると，図のような画面が表示されます．

（１）ボード（デバイス）の選択

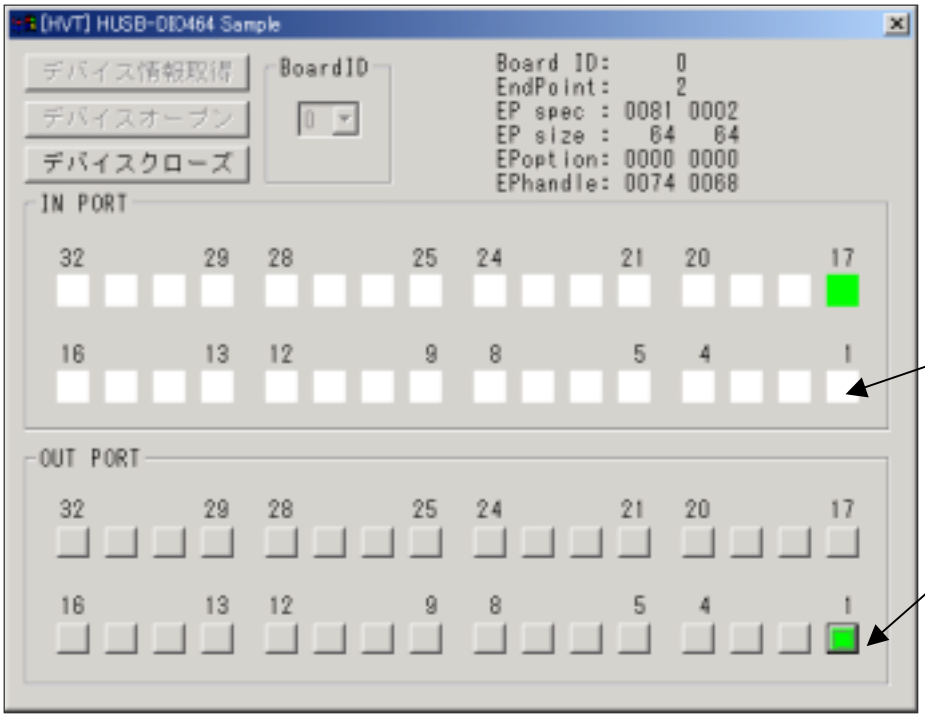
サンプルプログラムでは、ボード上の動作操作開始・終了は次の手順に従います。

①デバイス情報取得
②ボードの選択（２枚以上の場合）
③デバイスオープン
④デバイスクローズ
⑤サンプルプログラムの終了（デバイスクローズ後）

（注）④デバイスクローズ処理において、全出力ポート（１～４）には'０'を書込みます．

（２）ボード上の操作と表示

デバイスオープンを行いますと次の画面となります。



入力表示（読込）
入力状態と表示色
　入力中　・・緑
　入力なし・・白

出力ボタン（書込）
　ON/OFF 切替

出力表示（読込）
　出力状態と表示色
出力中　・・緑
出力なし・・なし（背景色）